République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : IA

Sujet :

Panneau de configuration et de contrôle des activités et services dans un intranet

Présenté par : BENTLEMSAN Khadidja
HASSAINE Nassima

Promoteur : Mr D. Bennouar

- 2006/2007-

Dédicaces

- *A une personne que j'aurais tant aimé qu'elle assiste à ma soutenance, mon très cher grand père Zoubir رحمه الله.*
- *A mes très chers grands parents : Mohammed, Hassina et Yamina.*
- *A mes très chers parents qui ont toujours été là pour moi, J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.*
- *A mes chers frères : Zoubir, Faiz, Ayoub et Younes.*
- *A mes chères tantes: Maouia, Nadia, Lotfia, Nacira, Fatma zohra, Houria, Naima, Nora et Faiza.*
- *A mon merveilleux cousin Lotfi.*
- *A toute la famille Bentlemsan et Boutouchent grands et petits*
- *A tous ceux qui me sont chers*

Je dédie ce travail

Khadidja

Dédicaces

A mon père — *Tes conseils, ton support et ta volonté m'ont toujours inspiré et sont pour moi la clé de ma réussite ;*

A ma mère — *Ton amour inconditionnel et ta grande charité forment la perle de mon existence. Tu es donc avec moi dans tous ce que je fais ;*

A ma sœur Salima — *Tes encouragements, tes judicieux conseils ont été des facteurs indispensables à la réussite de ce travail ;*

A mes chers frères Amine et Aymen ;

A toute ma famille grands et petits ;

A mes meilleurs amies : Bentlemsan Khadidja, Jaziri Fatma, Hassaine Yamina, Mancer Yassmine, Djellal Sarah, Dramsaoud Faiza, Dahimene Assya, Amrani Rim, Bensaad Souad, Fiak Afrah, Lekhal Nassima, Lamri Malika, Ziane Fatiha, Chine Baya, pour les bons moments qu'on a passé ensemble durant mes études.

Je vous dédie ce travail

Nassima HASSAINE

« Dans les sciences, le chemin est plus important que le but.

Les sciences n'ont pas de fin »

Erwin Chargaf

ملخص

تفرض الواجهة الغرافكية نفسها أكثر فأكثر تحت نظام لينكس باعتبارها طريقة سهلة و فعالة لكل مستخدم غير متعود على خفايا هذا النظام .

يتمثل الهدف الأساسي من هذا المشروع في تصميم و انجاز لائحة تحكم و مراقبة لمختلف الخدمات المتعلقة بالشبكة المحلية (مزود الويب، مزود ملفات، مزود تراسل و الجدار الناري) مقدمة بذلك يد عون معتبرة لمدير الشبكة و الذي لا يحتاج إلى تغير ملفات الإعدادات أو تنفيذ تعليمات الشيل.

Résumé

L'interface graphique sous linux s'impose de plus en plus comme solution simple et efficace pour tout utilisateur non familiarisé avec ce système.

L'objectif principal de notre travail est de concevoir et réaliser un panneau de configuration et de contrôle des différentes activités et services dans un réseau local qui sera une aide précieuse pour tout administrateur réseau sans avoir à éditer à la main les fichiers de configuration ou exécuter des commandes shell.

Abstract

The graphical interface under Linux is more and more establishing itself as a simple and effective solution for any user not familiar with this system.

The main objective of our work is to conceive and create a control panel of the various activities and services in a Local area network which will be a precious help for every network administrator without resorting to a manual editing of configuration files or executing shell commands.

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES FIGURES

Page

LISTE DES FIGURES

Page

LISTE DES FIGURES

Page

LISTE DES FIGURES

Page

SOMMAIRE

I - PRESENTATION DU SUJET

Linux est un système d'exploitation puissant, modulaire et évolutif bénéficiant des fonctionnalités et des utilitaires habituellement livrés avec les variantes commerciales d'Unix.

A l'origine, il n'était pas doté d'une interface graphique, elle est relativement récente, rajouté par le biais d'un serveur X et a permis l'élargissement de cercle des utilisateurs en rendant plus facile l'exploitation de ce système (user friendly).

Toutefois, certaines tâches administratives comme la mise en place d'un pare feu, la création d'un compte utilisateur ou configuration d'un serveur requièrent une bonne compréhension de la structure des fichiers de configuration et une excellente connaissance en mode commande.

La réalisation des outils de configuration graphique se révèle donc indispensable pour apporter de l'aide à ceux qui souhaitent s'équiper de serveurs performants mais qui n'ont pas nécessairement les compétences ou le temps pour les administrer pleinement.

Dans ce cadre, plusieurs outils ont vu le jour, mais leur prise en main est un peu délicate, ils présentent quelques anomalies et ils impliquent parfois des interventions en mode console.

Notre travail suit la même politique, il consiste à réaliser un outil de configuration des principaux serveurs réseaux (serveur Web, serveur de Proxy, filtre de Proxy, serveur de messagerie, Firewall et NAT), basé sur une interface simple et ergonomique et en gardant non seulement l'administrateur réseau loin des fichiers de configuration et leurs règles syntaxiques, mais aussi de l'interpréteur de commande pour mener à bien les tâches d'administration des plus simples aux plus complexes.

Pour répondre aux exigences d'un tel système, nous avons choisi Fedora Core 6 qui est une distribution gratuite et offrant les services nécessaires pour l'administration réseau.

Au cours de cette recherche, nous avons pris connaissance de la configuration de chaque service, que nous avons expliqué dans ce document. Ce dernier se compose de 4 chapitres organisé comme suit :

Le premier chapitre présente l'analyse et la spécification des besoins où nous allons donner une description sommaire de ce qui doit être fait par notre système.

Le second chapitre couvre tout ce qu'il y a à connaître pour administrer un réseau, nous présentons les services réseaux, leurs fichiers de configurations et les règles d'or à appliquer pour assurer le bon fonctionnement.

Le troisième chapitre est consacré à la conception du système, nous abordons les détails du travail effectué, ceci se fera à l'aide des diagrammes UML pour mieux visualiser la structure des modules du système, ainsi que leurs comportement.

Le quatrième et le dernier chapitre comporte la réalisation du projet où nous allons exposer notre système et son mode de fonctionnement, nous présentons également les problèmes rencontrés et les solutions proposées pour y remédier.

II – METHODOLOGIE DE CONCEPTION ET DE REALISATION

Etant donné que :

- L'objectif est un logiciel local.
- Destiné à être réalisé par deux étudiantes sans aucune expérience.
- Non destiné obligatoirement à être terminé par un autre groupe.
- Sans contrainte budgétaire.
- Avec contrainte de la nécessité de mise en place dans les plus bref délai d'un prototype fonctionnel.
- Du niveau des étudiants en terme de maîtrise des outils de programmation.
- De la petite expérience en terme de programmation Java.
- De la nécessité d'une période d'apprentissage des outils.
- D'une période nécessaire à la compréhension du problème.
- D'une période nécessaire à la détermination progressives des vrais besoins.

La méthodologie à suivre est basée sur la programmation intensive, itérative et progressive (incrémentale).

La méthodologie incite à commencer par les aspects les plus simples, les réaliser et les tester (Exemple : test de chaque service) et passer ensuite à la réalisation d'un autre aspect.

Ce dernier aspect pourra mettre en cause les aspects précédents. Dans ce contexte un réajustement des étapes précédentes est nécessaire. Il faut revenir en arrière pour refaire la conception /Réalisation.

Ce processus pratique permettra d'une part de maîtriser l'outil de programmation et d'avoir à chaque étape une version fonctionnelle d'une partie du logicielle. Cette méthodologie faite partie d'une famille émergente de processus dits "agiles", qui se démarquent des démarches traditionnelles en mettant l'accent sur le travail d'équipe et la réactivité. La méthode XP fait partie de cette famille et la méthodologie que nous suivons s'apparente sur beaucoup d'aspect à l'***XtremeProgramming*** Cette méthodologie se focalise sur la construction proprement dite du logiciel, en aval des phases préparatoires d'études d'opportunité ou de faisabilité.

Dans notre méthode, comme :

- Le client (maîtrise d'ouvrage) pilote lui-même le projet, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines). Le client dans notre cas est le promoteur principal du sujet, à savoir Mr Djamal BENNOUAR, CC au Dpt Informatique de l'USDB.
- L'équipe formée de deux étudiantes en phase de préparation de leur mémoire de fin d'étude, livre, très tôt dans le projet une première version du logiciel, et les livraisons de nouvelles versions s'enchaînent ensuite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements.
- L'équipe constituée des deux étudiantes, s'organise elle-même pour atteindre ses objectifs, en favorisant une collaboration maximale entre ses membres.
- L 'équipe doit mettre en place un ensemble de jeux d'essai ou doit mettre en place des tests automatiques pour toutes les fonctionnalités qu'elle développe, ce qui devrait garantit au produit un niveau de robustesse très élevé.
- Les développeurs améliorent sans cesse la structure interne du logiciel pour que les évolutions y restent faciles et rapides.

Chapitre 1 :

Analyse des besoins

Chapitre 1

Analyse des besoins

Introduction

Dans ce chapitre et afin de clarifier les besoins, nous allons donner une vision globale du comportement de notre système à l'aide de diagramme de cas d'utilisation d'UML.

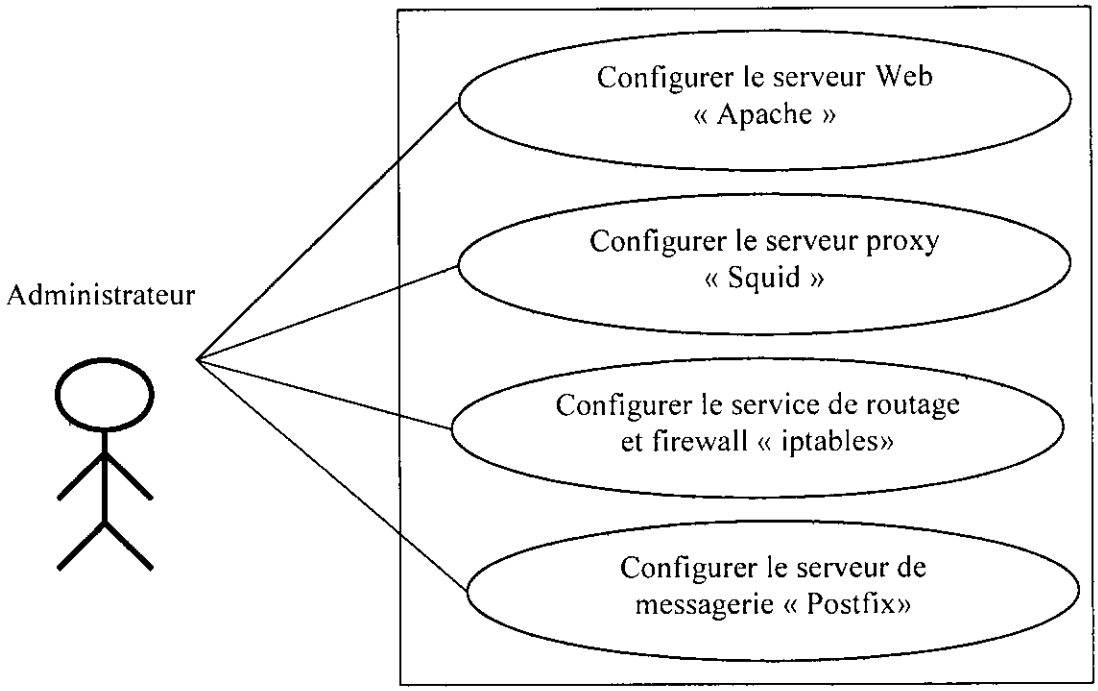
Un cas d'utilisation décrit un ensemble de séquence d'actions, y compris des variantes qu'un système exécute pour produire un résultat tangible pour un acteur [L_1].

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs du cas d'utilisation en interaction avec ces cas d'utilisation. En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système.

Dans la section suivante nous allons présenter les diagrammes de cas d'utilisation correspondant à chaque module présenté dans notre système.

I- Définition des besoins

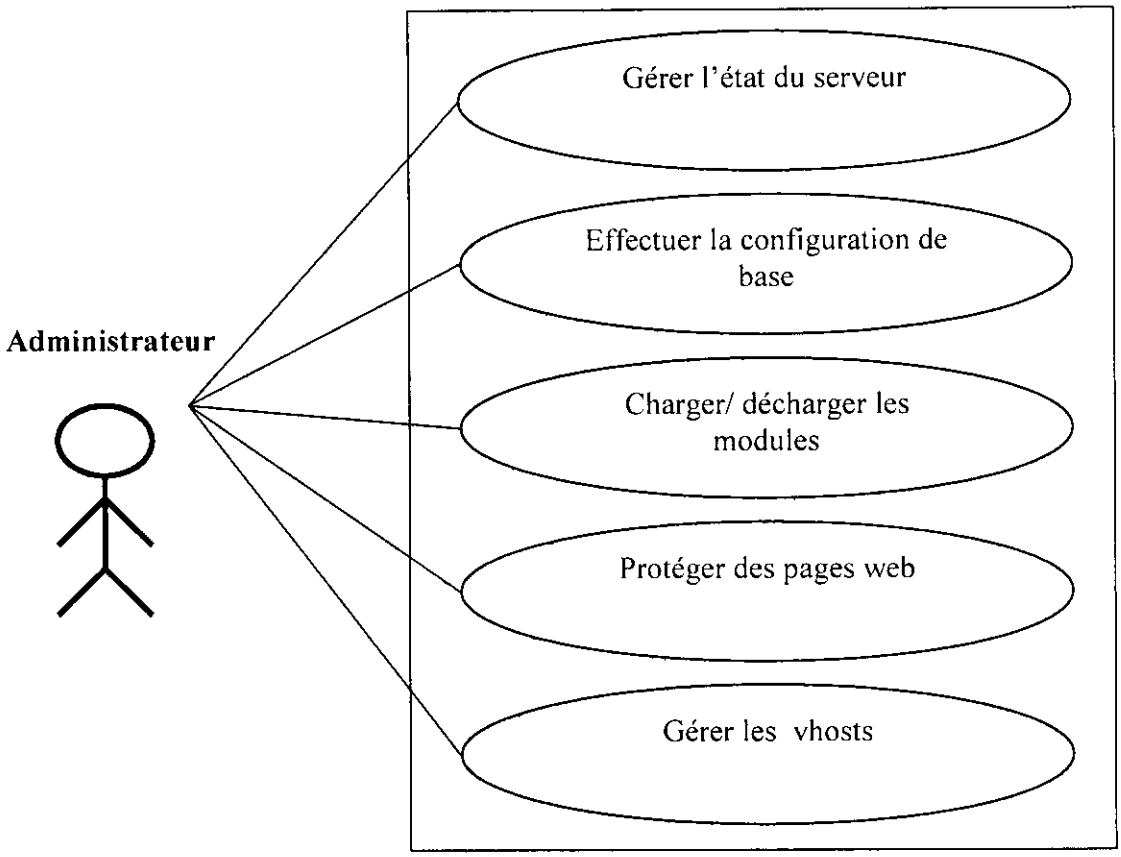
Dans notre cas, le diagramme de cas d'utilisation fait intervenir un seul type d'utilisateurs c'est l'administrateur réseau. Il s'organise en quatre modules fonctionnels, ceux introduits dans la *Figure 1.1*.



- Figure 1.1-

Dans ce que suit, nous allons détailler les cas d'utilisations correspondants à chaque module (éventuellement ses sous modules) :

I.1_Cas d'utilisation de la configuration du serveur web Apache :

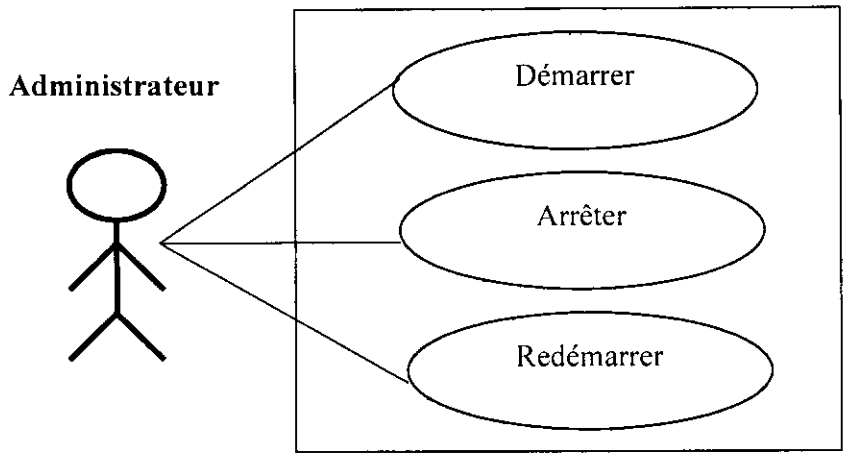


- Figure 1.2-

L'administrateur aura la possibilité d'effectuer une des tâches suivantes:

- Gérer l'état du serveur (voir *Figure 1.3*).
- Effectuer une configuration de base.
- Gérer les modules d'apache (voir *Figure 1.4*).
- Gérer les vhosts (voir *Figure 1.5*).
- Protéger les pages web.

- Cas d'utilisation pour la gestion du serveur :

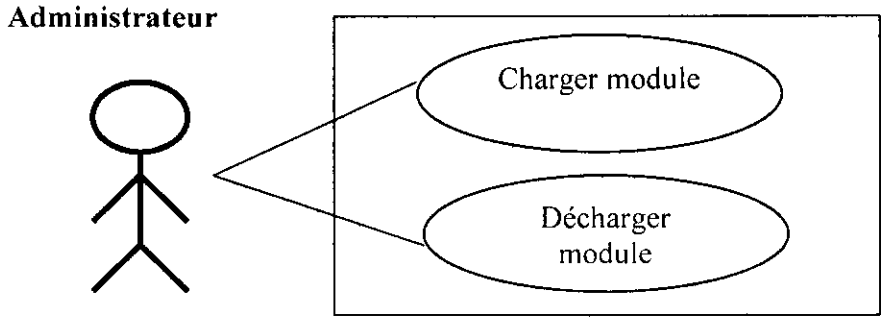


- Figure 1.3-

L'administrateur pourra à tout moment changer l'état du serveur, il peut donc :

- Démarrer le serveur.
- Arrêter le serveur.
- Redémarrer le serveur.

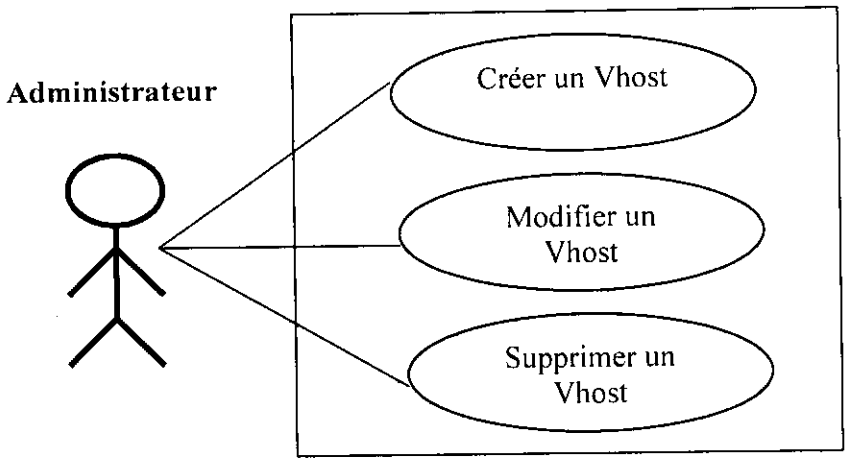
- Cas d'utilisation pour la gestion des modules :



- Figure 1.4-

L'administrateur pourra charger ou décharger un des modules d'apache.

- Cas d'utilisation pour la configuration des serveurs virtuels (vhosts) :

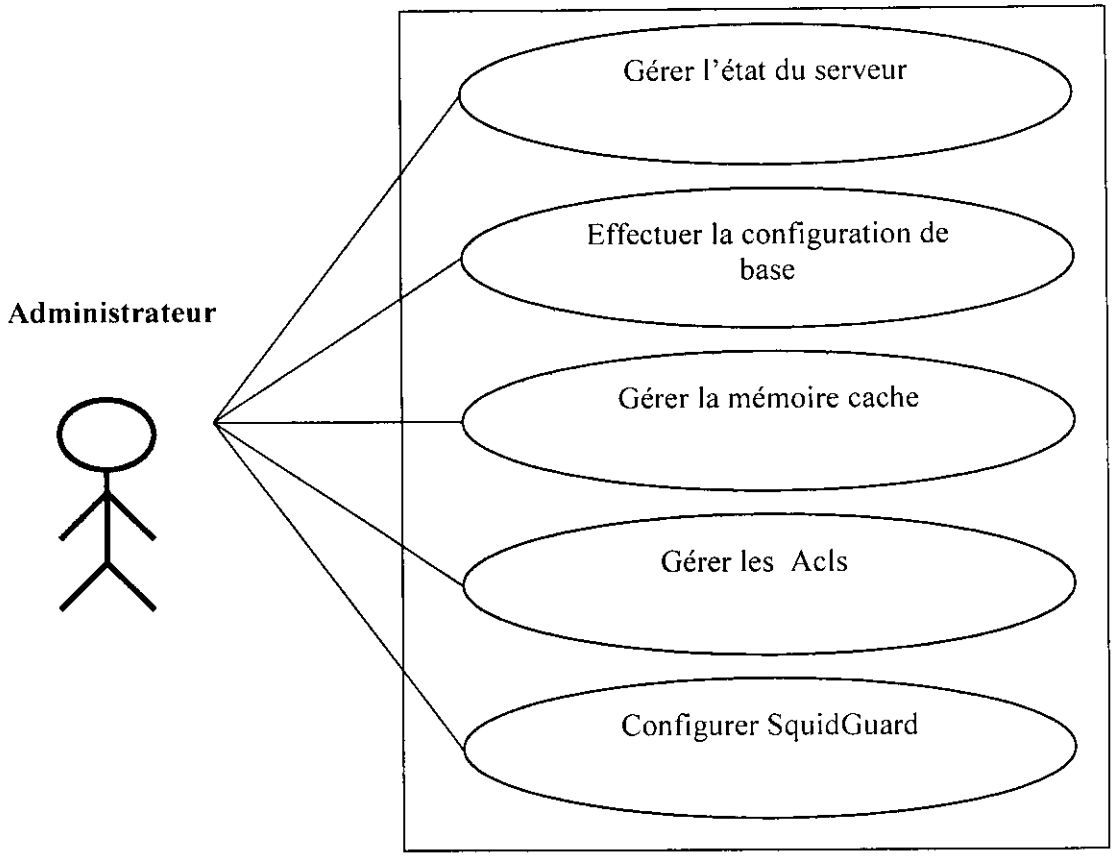


- Figure 1.5-

L'administrateur aura la possibilité de gérer les serveur virtuels il pourra :

- créer un serveur virtuel
- modifier les propriétés d'un serveur virtuel qui existe.
- Supprimer un serveur virtuel qui existe.

I.2_Cas d'utilisation de la configuration du serveur de proxy Squid :

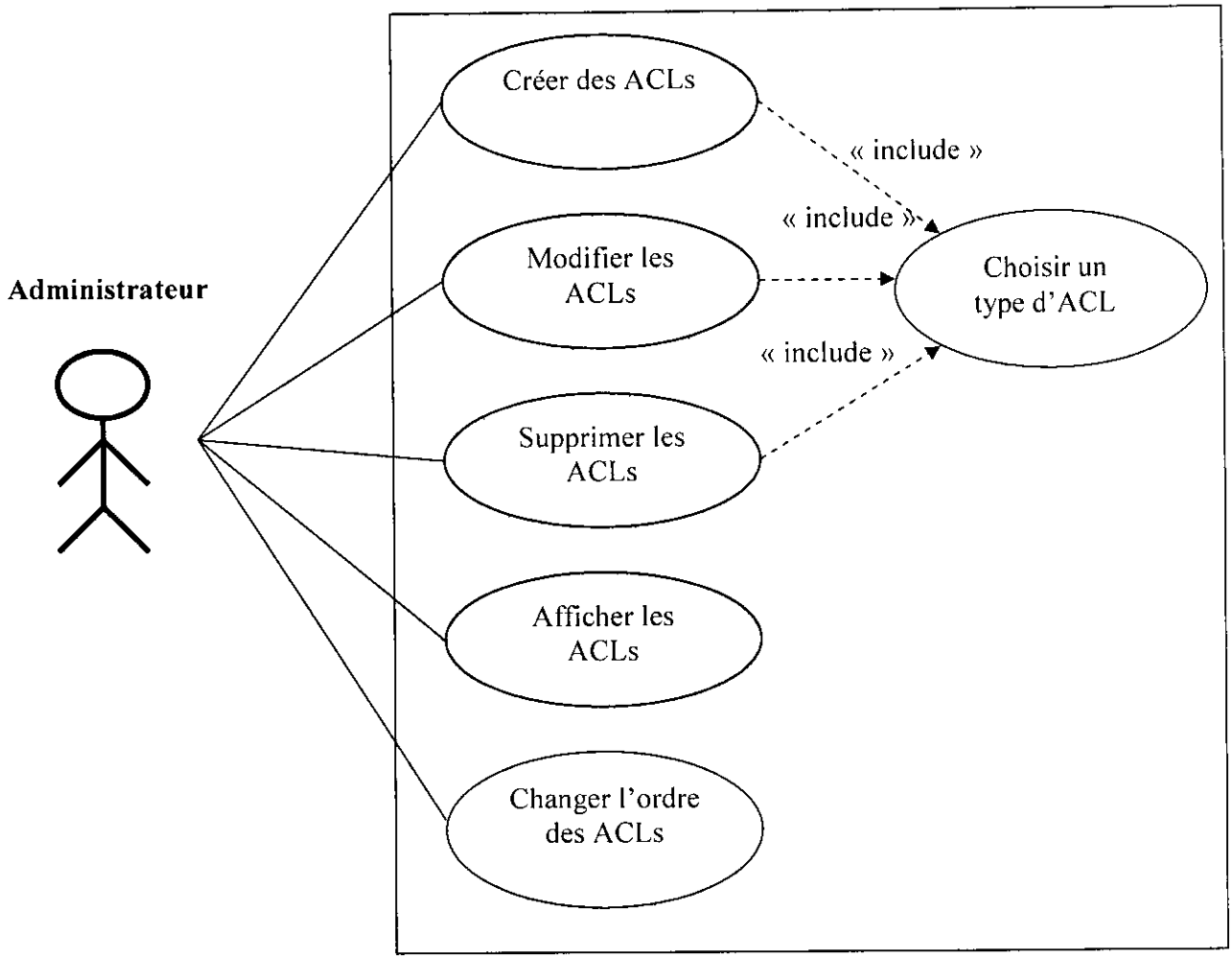


- Figure 1.6 -

Afin de gérer les ACLs l'administrateur pourra :

- Gérer l'état du serveur de la même manière que le serveur Apache (voir *Figure**1.3***).
- Effectuer une configuration de base.
- Gérer la mémoire cache (voir *Figure **1.8***).
- Gérer les acls (voir *Figure **1.7***).
- Configurer SquidGuard (voir *Figure **1.9***).

- Cas d'utilisation de la gestion des acls :

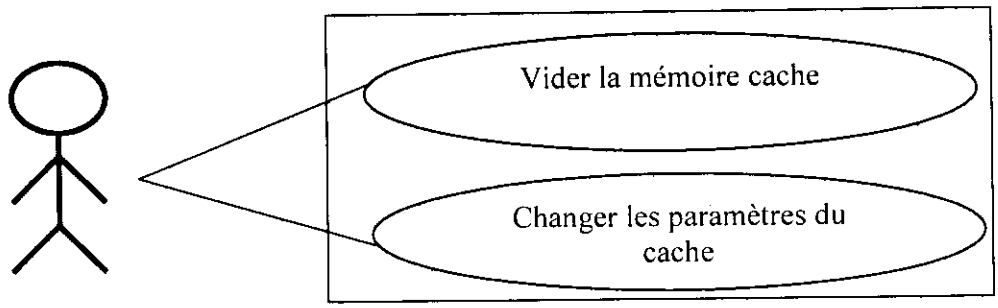


- Figure 1.7-

Dans le cas où l'administrateur souhaitera gérer une ACL, il pourra effectuer une tâche parmi les suivantes :

- choisir le type d'ACL désiré pour créer, modifier ou supprimer une ACL de ce type.
- Voir l'état des ACLs existantes.
- Changer l'ordre des ACLs existante.

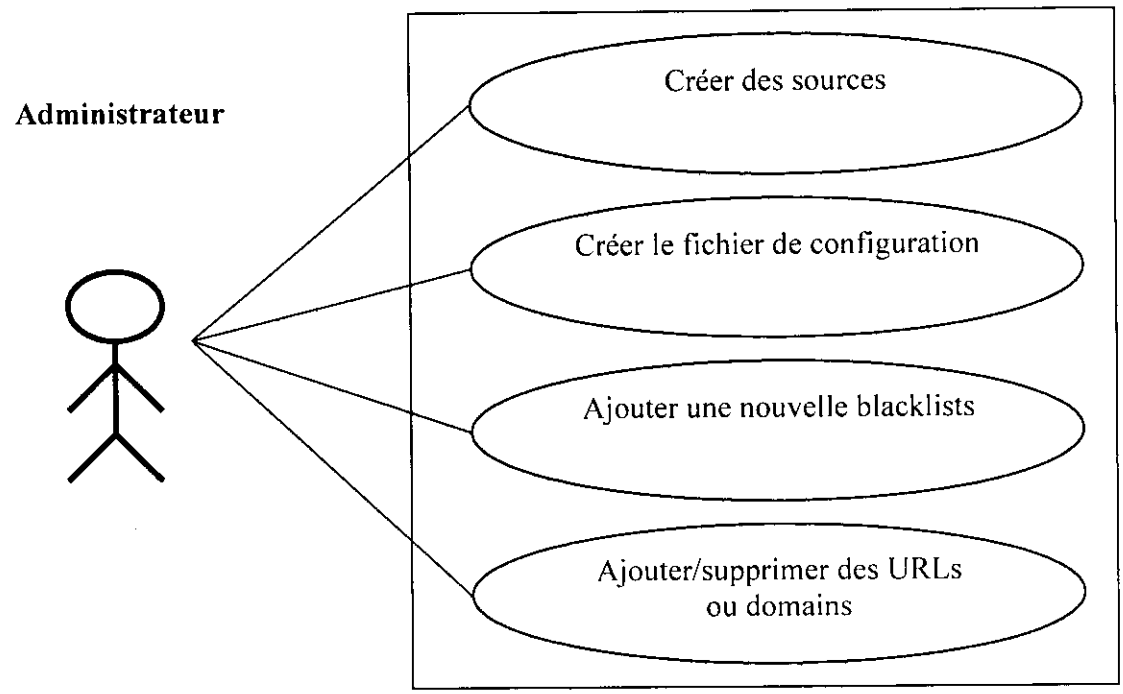
- Cas d'utilisation de la configuration de mémoire cache de squid :



- Figure 1.8-

Le système propose à l'administrateur de vider la mémoire cache de squid ou de changer les paramètres des objets en cache.

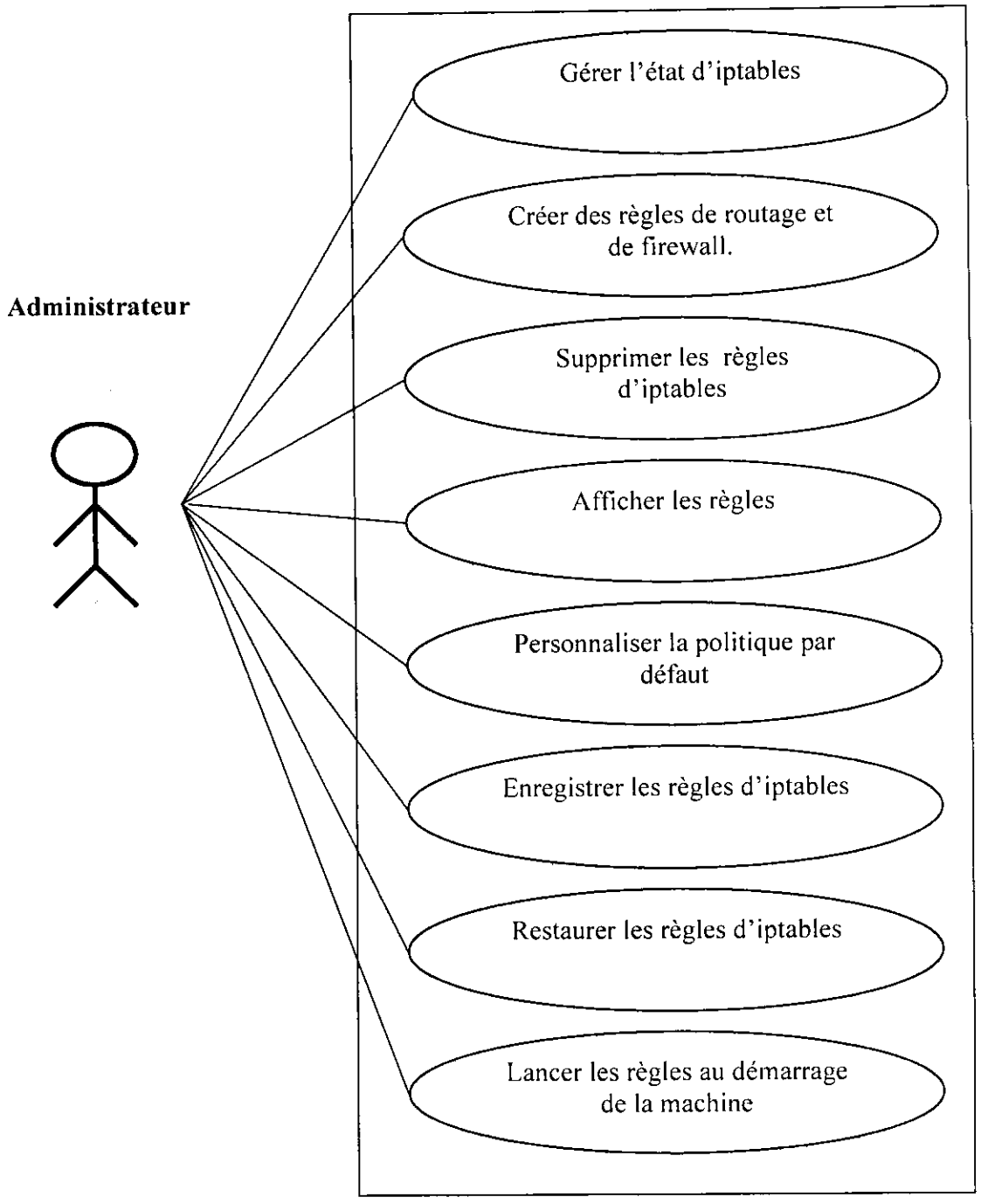
- Cas d'utilisation de la configuration du filtre SquidGuard :



- Figure 1.9-

Le système offre à l'administrateur la possibilité de créer des sources, créer un fichier de configuration, maintenir la blacklist en ajoutant une nouvelle catégorie ou en modifiant d'autres.

I.3_Cas d'utilisation de la configuration du service de routage et du Firewall :

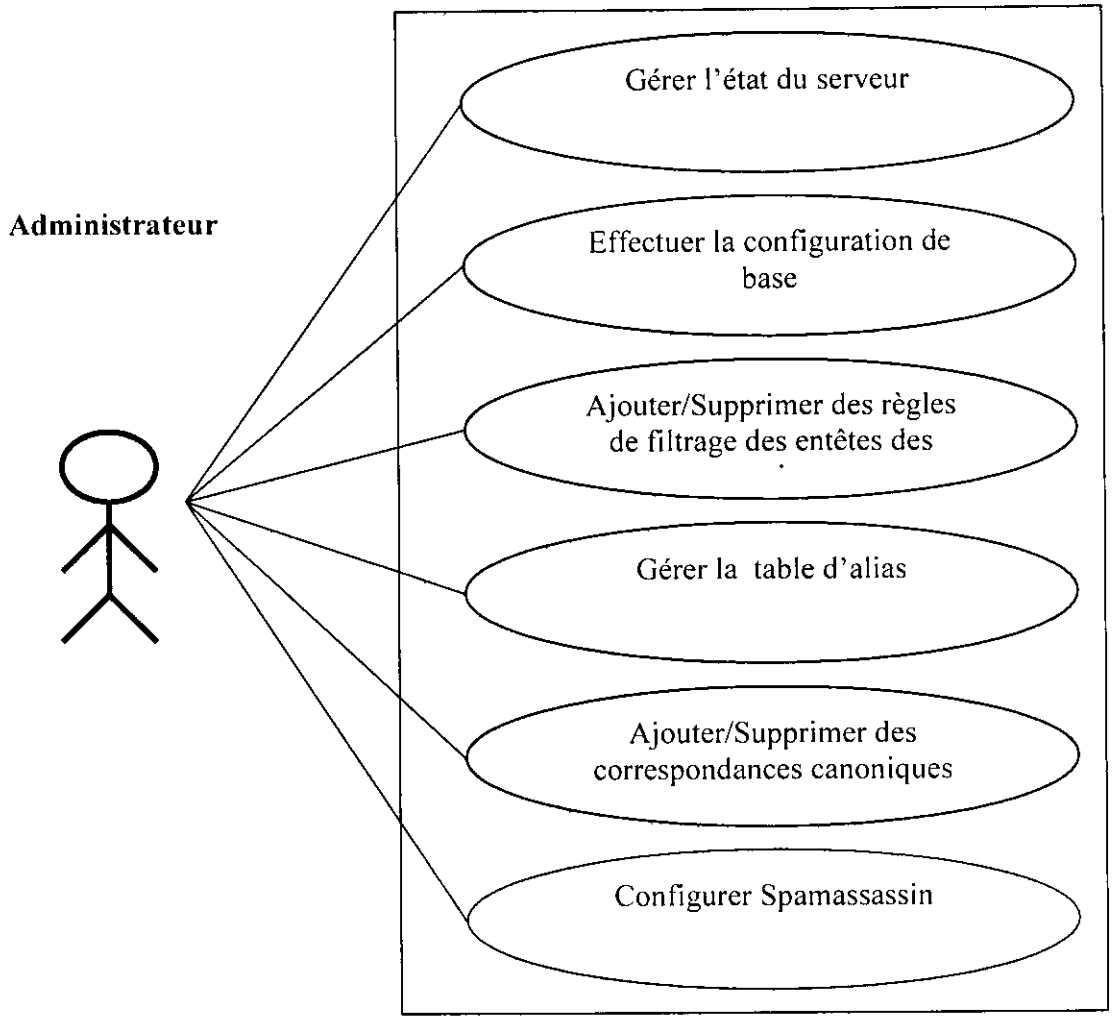


- Figure 1.10-

Pour gérer le service de routage et de Firewall l'administrateur pourra :

- Gérer l'état d'iptables (voir *Figure 1.3*).
- Créer les règles de routage et du firewall.
- Supprimer les règles existantes d'iptables.
- Afficher les règles d'iptables.
- Personnaliser la politique par défaut.
- Enregistrer les règles d'iptables.
- Restaurer les règles d'iptables.
- Lancer les règles d'iptables au démarrage de la machine.

I.4_Cas d'utilisation du serveur de messagerie :

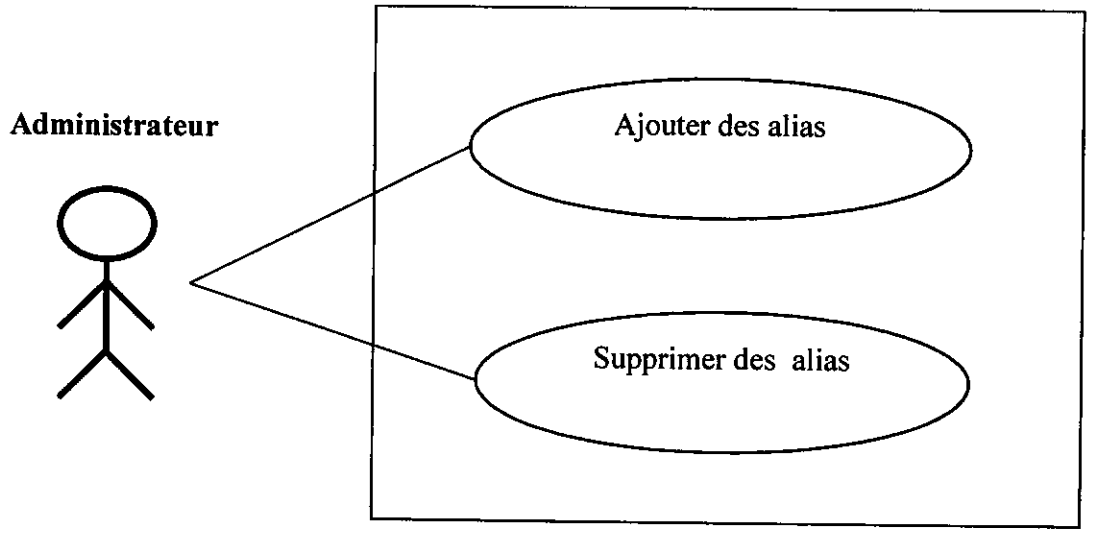


- Figure 1.11-

L'administrateur pourra :

- Gérer l'état du serveur (voir *Figure 1.3*).
- Effectuer la configuration de base de Postfix.
- Ajouter/supprimer des règles de filtrage des entêtes de courrier.
- Gérer la table des alias (voir *Figure 1.12*).
- Ajouter ou supprimer des correspondances canoniques.
- Configurer Spamassassin (voir *Figure 1.13*).

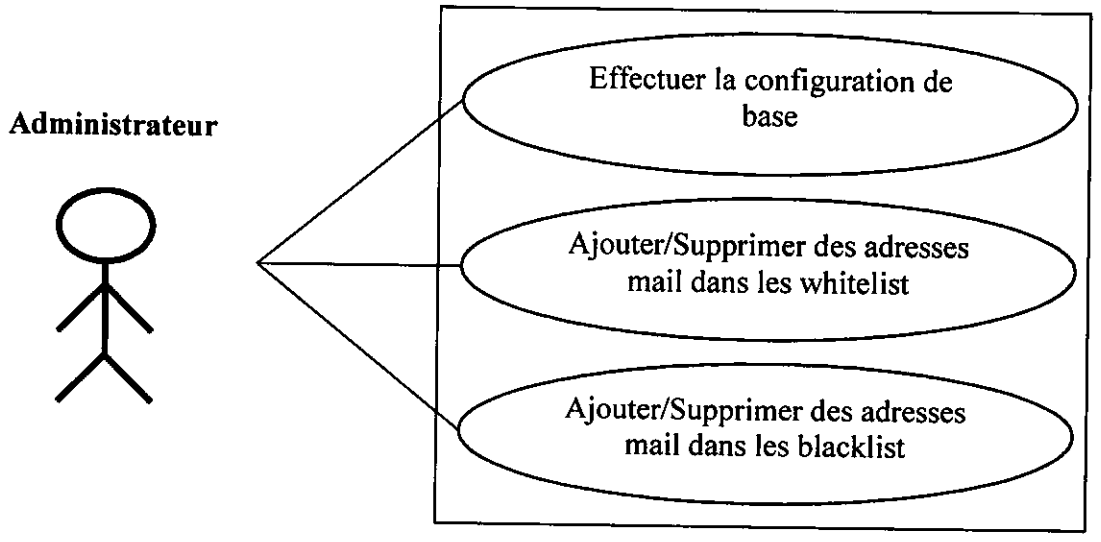
- Cas d'utilisation de la configuration de la table des alias :



- Figure 1.12-

L'administrateur pourra ajouter, modifier ou supprimer un alias.

- Cas d'utilisation du Spamassassin :



-Figure 1.13-

L'administrateur pourra effectuer une configuration de base, mettre à jour la blacklist (respectivement la whitelist).

Chapitre 2:

Etude des services réseaux

Chapitre 2

Etude des services réseaux à configurer

Introduction

Dans ce chapitre nous mettons l'accent sur les fichiers et les commandes qui se cachent derrière chaque service à configurer par notre application.

Nous commençons par la présentation du serveur Web Apache et son fichier de configuration, nous décrivons ainsi sa configuration de base et sa configuration avancée (création des serveurs virtuels, protection d'une page web).

Nous abordons ensuite le proxy Squid, sa configuration de base, configuration de mémoire cache et la création des listes d'accès (ACL).

Nous présentons par la suite le plugin SquidGuard, sa phase d'installation, sa configuration et la personnalisation de la liste noire.

Le service suivant concerne le routage et le Firewall, nous allons donc expliquer la structure d'Iptables (tables, règles) et nous illustrons par quelques exemples.

Enfin, la configuration du serveur de messagerie POSTFIX et celle de Spamassassin.

Serveur Web (Apache)

I.1.Introduction

Un **serveur** HTTP, est un logiciel servant des requêtes respectant le protocole de communication client-serveur Hypertext Text Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme « serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole **[WWW_1].**

Les principaux serveurs web sur le marché sont entre autres :

- Apache HTTP Server de la Apache Software Foundation, successeur du NCSA httpd.
- Microsoft IIS (Internet Information Server)
- Microsoft PWS (Personal Web Server)
- Xitami

Apache HTTP Server est le serveur HTTP le plus utilisé, qui sert environ 60% des sites Web en 2007 selon Netcraft.

Caractéristiques d'Apache:

- Logiciel gratuit.
- Code source disponible et modifiable permet un développement rapide du serveur, la création de modules spécifiques et une très grande réactivité dans la correction de tout bug identifié.
- Très grande flexibilité du serveur grâce à sa structure modulaire l'ajout d'un nouveau module permet d'ajouter de nouvelles fonctionnalités.

I.2.Type de matériel pour un serveur Apache sous Linux :

Apache s'exécute sur n'importe quel type de machine. Pour un serveur de sites WEB peu exigeants, un simple 486 fera parfaitement l'affaire. Pour des sites très exigeants utilisant de nombreuses bases de données, un Pentium multiprocesseur peut être envisagé.

- Concernant la mémoire :

Plus il y a de mémoire vive et plus la quantité de donnée en mémoire est importante ce qui a pour conséquence d'accélérer les accès.

- Concernant le disque dur :

Un disque dur rapide permet d'améliorer les performances d'accès aux données des sites WEB. A noter que dans le cas de sites à grande audience, il est préférable d'utiliser plusieurs disques de tailles moyennes plutôt qu'un seul disque à grande capacité (un disque dur ne pouvant lire qu'à un seul endroit à la fois).

- Concernant la carte réseau :

Une carte Ethernet 100baseT est préférable à une carte 10baseT. Si le serveur doit également être connecté à un Intranet local, on peut envisager d'utiliser deux cartes réseaux : une dédiée au réseau interne et l'autre au réseau Internet. Ce qui permet d'augmenter la sécurité et d'éviter que l'encombrement du serveur WEB diminue la bande passante du réseau interne.

I.3.Configuration du serveur Apache :

La configuration du serveur Apache s'effectue en modifiant son fichier de configuration. Le fichier de configuration d'Apache se nomme `httpd.conf` et est placé dans le répertoire /etc/httpd/conf , constitué de dizaines de lignes contenant des directives de configuration, leurs paramètres, leurs options ainsi que leurs commentaires.

Voici un extrait du fichier de configuration :

```
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2/> for detailed information.
# In particular, see
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin root@localhost

#
# ServerName gives the name and port that the server uses to identify
itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
ServerName localhost
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"
#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to deliver content-
# negotiated documents.  The MultiViews Option can be used for the
# same purpose, but it is much slower.
#
DirectoryIndex index.html
```

I.3.1.Configuration de base du serveur Apache :

Le tableau ci-dessous représente quelques directives de configuration de base, leurs valeurs par défaut ainsi que leurs descriptions :

Directive	Valeur par défaut	Description
ServerName	localhost	Le nom avec lequel apache envoie sa réponse.
DocumentRoot	/var/www/html	Emplacement de pages web.
DirectoryIndex	Index.html	Page par défaut.
ServerRoot	/etc/httpd	Emplacement du fichier de configuration.
ServerAdmin	root@localhost	Adresse électronique de l'administrateur

Tableau 1 : Description de quelques directives de configuration d'Apache

I.3.2.Les serveurs virtuels

Par défaut, on ne fait tourner qu'un seul site web sur un serveur. Mais il peut parfois être intéressant de disposer de plusieurs noms qui pointent chacun sur des répertoires différents (Un utilisateur pour un même serveur apache croira avoir plusieurs).

Exemple :

Pour mettre en place un autre serveur (dans cet exemple test.dz qui a ses pages dans le répertoire `/var/www/html/NewFolder`), on éditera le fichier httpd.conf dans le répertoire de configuration de Apache (/etc/httpd/conf), pour y ajouter les lignes suivantes :

```
############################    DEBUT DES VHOST    ######################
NameVirtualHost 192.168.0.1:80

############################    LE VRAI SERVEUR    ######################

<VirtualHost  localhost:80>
ServerAdmin root@localhost
ServerName localhost
DocumentRoot "/var/www/html"
DirectoryIndex index.html
ErrorLog logs/error-log
</VirtualHost>

############################    LE PREMIER VHOST    ######################

<VirtualHost  test.dz:80>
ServerAdmin vhost@localhost
ServerName test.dz
DocumentRoot /var/www/html/NewFolder
DirectoryIndex mapage.html
ErrorLog logs/error-test.dz-log
</VirtualHost>

############################    FIN DES VHOST    ######################
```

La seconde étape est d'ajouter la ligne suivante dans le fichier hosts « /etc/hosts » :

```
test.dz  <ADRESSE IP DU SERVEUR >
```

Note

Le serveur principal doit être mis en premier car c'est lui qui sera activé sur un simple appel de la machine par son adresse IP.

Description des paramètres de vhosts :

NameVirtualHost	Adresse IP	
<VirtualHost Nom_vhost>	Début de la configuration	
ServerAdmin	email@domain.com	Cette adresse peut être différente de celle du vrai serveur (localhost).
DocumentRoot	Chemin	le chemin complet jusqu'au répertoire du site (ce n'est pas forcement le même que celui du vrai serveur).
ServerName	Nom du vhost	le nom du VirtualHost comme nom de Serveur.
ErrorLog	Log / fichier log du vhost	Pour plus de lisibilité, on stocke les erreurs dans un autre fichier que celui du vrai serveur.
< /VirtualHost >	Fermer la configuration de vhost.	

Tableau 2 : Description de quelques paramètres du vhost.

Note

Après chaque modification du fichier de configuration d'apache (httpd.conf) le serveur Apache doit redémarrer avec la commande Shell « service httpd restart» pour la prise en charge des nouvelles modifications.

I.3.3.Protection d'un répertoire :

Il est parfois impératif de protéger certains répertoires du serveur pour éviter que n'importe qui puisse y accéder. Que ce soit par Internet ou en Intranet, il y a différentes méthodes de procéder.

La méthode la plus facile à gérer est celle de .htaccess qui s'applique à un répertoire et à tous ses sous répertoires, mais peuvent être modifiés par un autre fichier .htaccess dans un sous répertoire.

Un fichier .htaccess est un fichier texte simple contenant des commandes Apache comme celles-ci :

```
AuthUserFile password/.htpasswd
AuthGroupFile /dev/null
AuthName "Identification"
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

Quelques explications

- **AuthUserFile** : C'est le chemin d'accès au fichier qui contiendra les mots de passe. S'il ne commence pas par un slash (/), ce sera un sous répertoire du serveur web. Ici, si les pages web sont dans le répertoire **/var/www/html**, les mots de passe seront dans **/var/www/html/password/.htpasswd**.,mais il vaudra mieux placer ce fichier en dehors du site web pour que personne ne puisse y accéder.
- **AuthGroupFile** : pointe toujours vers /dev/null. Il faut que cette ligne soit présente.
- **AuthName** : c'est le texte qui apparaîtra dans la fenêtre demandant les mots de passe.
- **AuthType** : L'authentification est en générale « basic ». Les mots de passe sont alors envoyés en clair sur le réseau.
- **Limit** : C'est ici qu'on va indiquer ce qui est autorisé et interdit dans le répertoire. Les commandes GET et POST indiquent la récupération de pages web et la réponse à certains formulaires.

- **Require :** On peut entrer ici « valid-user », ce qui accepte tous les utilisateurs qui ont un login : mot de passe dans .htpasswd.

Génération des mots de passe :

Pour créer le fichier .htpasswd sous Linux, on utilise la commande htpasswd sous la forme suivante :

```
htpasswd -c .htpasswd username
```

et cela ajoutera au fichier .htpasswd de ce répertoire une ligne de ce type :

username: v3l0KWx6v8mQM

Il est aussi possible d'utiliser des générateurs de mots de passes en ligne comme :

- Raptor.golden.net : **[WWW_9]**
- htpasswd file generator : **[WWW_10]**

Serveur de proxy (SQUID)

II.1.Introduction :

Un serveur "proxy" permet de partager un accès Internet entre plusieurs utilisateurs avec une seule connexion. Un bon serveur proxy propose également un mécanisme de cache des requêtes, qui permet d'accéder aux données en utilisant les ressources locales au lieu du web, réduisant les temps d'accès et la bande passante consommée.

Squid est un logiciel de cette catégorie, qui autorise le proxy, le cache des protocoles HTTP, FTP, SSL et Gopher. De ce fait il permet d'accélérer les connexions à l'internet en plaçant en cache les sites les plus visités. Il supporte également les contrôles d'accès et fournit une trace complète (log) de toutes les requêtes.

- **Où** trouver Squid ?

Squid est disponible sur le cd de toutes les distributions de Linux, il suffit juste de le cocher lors de l'installation. Il est aussi téléchargeable sur le site officiel **[WWW_2]**.sous forme de fichiers tar ou rpm.

Plate forme matérielle :

Les ressources d'UC nécessaires ne sont pas élevées pour SQUID. Par exemple un vieux pentium III avec un disque de 10 Go et une RAM de 128 Mo fera parfaitement l'affaire.

Plate forme logicielle :

SQUID peut tourner sur bon nombre de système d'exploitation

- Linux.
- FreeBSD, NetBSD.
- OSF/ Dec Unix.
- AIX.
- Windows NT.

II.2.Configuration de Squid :

Le fonctionnement et le comportement de Squid sont gérés par les indications de configuration figurant dans le fichier squid.conf, ce fichier est en général placé dans le répertoire /etc/squid, les paramètres par défaut lui permettent de fonctionner, mais pas pour bloquer ou autoriser des machines.

II.2.1.Configuration de base :

Le tableau suivant résume la signification des principales options ainsi que leurs valeurs par défaut

- Les ports :

Directive	Valeur par défaut	Description
http_port	3128	le numéro de port que Squid va écouter pour satisfaire les requêtes des clients.
Icp_port	3130	c'est le port sur lequel le cache peut être interrogé par un cache fils ou voisin, cas d'une hiérarchie.

Tableau 3 : Les ports de squid.

- Les paramètres d'administration :

cache_mgr	root@kouba.dz	Permet de spécifier l'adresse Email de l'administrateur du serveur. Ainsi, en cas d'arrêt de celui-ci un e-mail lui sera envoyé.
visible_hostname	root	c'est le nom que renvoi squid lorsqu'il est interrogé de l'extérieur.
error_directrory	error_directory /usr/lib/squid/errors/English	Répertoire des messages d'erreur.

Tableau 4 : Description de quelques directives de base de squid.

- Le temps d'attente :

Il est nécessaire d'établir des délais au delà desquels Squid considérera qu'une transaction a échoué. Ceci vaut pour l'établissement des connexions, l'envoi des requêtes et la réception des résultats.

Directive	Valeur par défaut	Description
connect_timeout (secondes)	30	le temps d'attente d'une réponse du serveur distant avant de retourner une page de type « connection_Timeout » au client.
request_timeout (minutes)	5	le temps d'attente de Squid entre deux requêtes HTTP avant de fermer la connexion.
pconn_timeout (secondes)	60	le temps d'attente de Squid avant de fermer une connexion de type persistante

Tableau 5 : Le temps d'attente de squid.

II.2.2.Le cache :

Le cache est bien plus qu'un service rendu par Squid à vrai dire. Il fait partie intégrante de Squid et justifie à lui seul l'utilisation de Squid pour un réseau partageant un même accès à Internet. Le rôle d'un serveur cache est de stocker les objets demandé par les utilisateurs pour la première fois via les protocoles HTTP, FTP et Gopher. Ainsi, lors des demandes futures sur un objet présent en cache, le serveur de cache n'aura pas besoin d'aller chercher cet objet sur internet et retournera directement celui qu'il a en mémoire. Les objets stockés peuvent être de tout type : texte, image, vidéo, ...

Ce mécanisme procure deux grands avantages non négligeables:

- Une économie de bande passante d'autant plus grande que le volume de données représentées par l'objet. Le nombre de requête qui pourront être détournées vers le cache dépend de plusieurs paramètres comme la durée d'activité du cache, la durée de vie des objets dans le cache, le nombre de clients et de requêtes sur le réseau, etc.
- Un gain de temps pour les utilisateurs internet. Effectivement lorsque le cache prend en charge une requête, le délai d'accès à un objet devient celui de la recherche de cet objet dans le cache, plus les temps d'accès réseau.

Les paramètres du cache :

- Réinitialisation du cache :

Il arrive parfois que le cache local n'ait pas la bonne forme, car squid n'a pas été arrêté correctement. Il faut donc le reconstruire, par l'intermédiaire des deux commandes suivante :

```
rm -rf /var/spool/squid/*
squid -z
```

- Les autres paramètres :

Squid ne stocke les objets sur disque que si leur taille entre dans un intervalle fixé par l'administrateur, de même pour le nombre d'adresses IP stockées.

Directive	Valeur par défaut	Description
maximum_object_size (kO)	1024	Permet de spécifier la taille maximale des objets qui seront stockés dans le cache.
minimum_object_size (KO)	0	Permet de spécifier la taille minimale des objets qui seront stockés dans le cache.
ipcache_size (Adresses)	1024	Permet de spécifier le nombre d'adresses IP qui seront enregistrés.

Les documents cachés seront stockés dans une arborescence de racine cache_dir , La première constante indique la taille maximale du cache en Mo. Ces derniers seront stockés dans des répertoires répartis sur deux niveaux. Une fonction de hachage garantit une répartition équitable des objets.

cache_dir	< chemin > <Taille_cache><level1><level2> `cache_dir ufs /usr/local/squid/cache 1024 16 256`	pour indiquer le répertoire devant accueillir les objets, la taille disque, le nombre de répertoires de premier et de second niveau pour le stockage.

Tableau 6 : paramètres du cache.

II.2.3.Contrôle d'accès :

Par défaut squid n'autorise personne à accéder au cache. Il faut donc donner les droits nécessaires pour permettre aux machines désirant d'accéder au cache. La gestion de la sécurité se fait par l'intermédiaire de ce que l'on appelle des ACL (abréviation de l'anglais « Access Control List »).

Les ACLs permettent non seulement d'autoriser l'accès au cache, mais elles servent aussi au filtrage, comme interdire l'accès à certains sites, interdire le cache de certain type de page, interdire l'accès à certaines URLs .....etc.

Squid dispose pour ces contrôles de deux types de composants : les éléments ACL et la liste d'accès. Une liste d'accès, autorise ou refuse l'accès au service.
Ci-dessous quelques uns des plus importants éléments ACL :

src	Source, c-à-d l'adresse ip du client, il est possible d'utiliser une plage d'adresse sous la forme : **adresse_ip_debut-adresse_ip_fin**
dst	Destination, c-à-d l'adresse ip de l'ordinateur cible
srcdomain	Source, c-à-d le nom de domaine du client
dstdomain	Destination, c-à-d le nom de domaine de la destination
time	Heure du jour et jours de la semaine
url_regex	Expression régulière décrivant une catégorie d'URL
urlpath_regex	Expression régulière décrivant un ensemble d'URL sans le protocole ni le nom d'hôte.

Tableau 7 : Les éléments d'ACL.

Pour activer le contrôle il faut d'abord définir un ensemble d'ACL et ensuite y appliquer des règles. Le format d'une ACL suit la syntaxe :

acl acl_element_name type_of_acl_element values_to_acl

Note :

1) acl_element_name peut être n'importe quel nom attribué par l'utilisateur à un élément ACL.
2) Deux éléments distincts ne peuvent avoir le même nom
3) Chaque ACL est une liste de valeurs. Pendant la vérification, les valeurs multiples utilisent un OU logique. Autrement dit un élément ACL correspond si l'une des valeurs est reconnue.

Différentes listes d'accès sont disponibles. Celles que nous avons utilisées sont décrites ci-dessous :

http_access	Autorise les clients HTTP à accéder au port HTTP. C'est l'ACL primaire.
cache	Définit le cache pour les réponses aux requêtes

Une règle de liste d'accès comporte les mots **allow** ou **deny**, ce qui autorise ou refuse un service pour un élément ACL particulier ou pour un groupe d'éléments.

Note

1) Les règles sont vérifiées dans l'ordre où elles ont été écrites et se terminent dès qu'une correspondance a été établie.
2) Une ACL peut comporter plusieurs règles.
3) Si aucune correspondance n'est trouvée, l'action par défaut est l'inverse de la dernière règle de la liste, il est donc préférable d'être explicite sur l'action par défaut.

4) Tous les éléments d'une même entrée d'accès sont associés par un ET s'exécutant de la manière suivante :

http_access Action statement1 AND (ET) statement2 AND (ET) statement OR (OU).
http_access Action statement3

Des règles multiples de http_access sont comparées par des OU alors que les règles d'une entrée d'accès sont associées par des ET.

5) Les règles sont lues de haut en bas.

II.2.3.1.Quelques exemples d'ACL dans le fichier squid.conf

Exemple 1 :

```
acl QUERY urlpath_regex cgi-bin \?
cache deny QUERY
```

Explication

La règle **cache deny QUERY** ci-dessus permet d'interdire le cache pour les pages de type .cgi et ?.

Exemple 2 :

```
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
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
http_access deny !Safe_ports
```

Explication

La règle **http_access deny !Safe_ports** permet de refuser l'accès des ports qui ne sont pas définis dans l'acl Safe_port.

Exemple 3:

```
acl bad_domain dstdomain -i "/etc/squid/baddomain.txt"
http_access deny bad_domain
acl Filtrer_url url_regex -i "/etc/squid/URL.txt"
http_access deny Filtrer_url
acl url_mp3 url_regex -i \.mp3
http_access deny url_mp3
acl url_wav url_regex -i \.wav
http_access deny url_wav
acl url_swf url_regex -i \.swf
http_access deny url_swf
### ACL ###
### ACL3 ###
acl ACL3 src 192.168.0.1
acl temps_ACL3 time ASMTW 7:30-15:00
http_access allow ACL3 temps_ACL3
# # # FIN ACL3 # # #
### ACL2 ###
acl ACL2 srcdomain projet.ens-kouba.dz
acl temps_ACL2 time  8:00-16:00
http_access allow ACL2 temps_ACL2
# # # FIN ACL2 # # #
### ACL1 ###
acl ACL1 src 192.168.0.100-192.168.0.200
http_access allow ACL1
# # # FIN ACL1 # # #
http_access deny all
```

Explications

La règle **http_access deny bad_domain** permet d'interdire l'accès aux domaines qui sont écrits dans le fichier baddomain.txt

- La règle **http_access deny Filtrer_url** permet d'interdire l'accès aux URL contenant les mots clés qui sont dans le fichier URL.txt, et aux URLs qui sont écrites dans le fichier URL.txt
- Les règles **http_access deny url_mp3**, **http_access deny url_wav**, et **http_access deny url_swf**, permettent respectivement d'interdire l'accès aux extensions .mp3, .wav et .swf .
- La règle **http_access allow ACL3 temps_ACL3** autorise la machine, dont son adresse ip est 192.168.0.1, à accéder au cache de 7 :30 à 15 :00, dans les jours de semaine : Samedi, Dimanche, Lundi, Mardi, et Mercredi.
- La règle **http_access allow ACL2 temps_ACL2** autorise la machine dont son nom de domaine est : projet.ens-kouba.dz à accéder au cache de 8 :00 à 16 :00.
- La règle **http_access allow ACL1** : Autorise toutes les machines dont leurs adresses ip appartiennent à la plage d'adresse : 192.168.0.100-192.168.0.200 à accéder au cache.
- A la fin il y a la règle **http_access deny all** c'est la politique par défaut.

Enfin, Il ne reste qu'à lancer Squid en invoquant la commande :

```
# service squid start
```

Il faudra ensuite configurer le Navigateur Web (sur toute machine choisie pour permettre l'accès à Squid) pour qu'il utilise Squid comme serveur proxy.

Filtre de proxy (SQUIDGUARD)

III. 1. Introduction

SquidGuard est un plugin de Squid largement utilisé, qui utilise la librairie Berkeley Database de sleepycat. Créé par « Päl Baltzersen » et « Lars-Erik Häland », Il offre les possibilités suivantes :

- Filtrage.
- Redirection.

Il s'agit en outre d'un logiciel libre (sous licence **GPL**), portable (disponible sur la majorité des systèmes **Unix**) **[WWW_3]**.

Utilité de SquidGuard

On pourra utiliser SquidGuard pour ces tâches :

- Limiter l'accès à certains sites Web.
- Bloquer l'accès à certaines url.
- Rediriger les bannières sur des fichiers GIF vides.
- Avoir des règles basées sur des dates ou sur des groupes

III. 2. Installation

Fedora Core est cousine de RedHat et utilise logiquement les RPM (RPM Package Manager) qui sont des archives contenant des programmes précompilés prêts à l'emploi. Evidemment, il est possible comme sur certains systèmes non communautaires de les chercher un par un sur internet et de les installer manuellement. Mais la gestion des dépendances peut quelquefois devenir un véritable casse-tête.YUM est un outil permettant de gérer les installations, les désinstallations et les mises à jour de paquetages au format RPM. Il gère les dépendances en téléchargeant ce qui est nécessaire. Il trouve les paquetages sur différentes sources (site internet) que l'on appelle des dépôts.YUM est fourni en standard dans toutes les versions de Fedora Core. Donc pour installer SquidGuard il suffit de taper en root :

```
yum install squidGuard
```

III. 3. Configuration de SquidGuard

La configuration de SquidGuard est réalisée dans le fichier squidguard.conf. Il est en général installé dans le répertoire /etc/squid/squidguard.conf. Et il a une structure bien précise.

Nous allons présenter un fichier squidGuard.conf accompagné des commentaires et des explications des différentes parties du fichier.

```
# le fichier de conf de SquidGuard
dbhome /var/squidGuard/blacklists
logdir /var/log/squidGuard

time afterwork {
  weekly    * 00:00-08:00
  weekly    * 17:00-24:00
  weekly    fridays 16:00-17:00
  weekly    saturdays sundays
  date      *.01.01
  date      *.11.01
  date      2006.04.14-2006.04.17
  }
src admins {
        ip      192.168.2.0-192.168.2.255
        ip      172.16.12.0

}


dest ads {
        domainlist ads/domains
        urllist ads/urls
        log ads
}
dest mail {
        domainlist mail/domains
        urllist mail/urls
        log mail
}
acl {
        admins within afterwork {
                pass all
        }
        else    {
                pass    !ads !mail all
                redirect http://ser/t.htm
                }
        default {
                pass    none

                }
}
```

Chemins d'accès :

dbhome /var/squidGuard/blacklists	répertoire de la blacklist
logdir /var/log/squidGuard	Répertoire des logs

Plages horaires :

`time` afterwork {	définit une plage
weekly * 00:00-08:00	toutes les nuits
weekly * 17:00-24:00	toutes les soirées
weekly fridays 16:00-17:00	tous les vendredis de WE
weekly saturdays sundays	tous les samedis et dimanches
date *.01.01	tous les premiers de l'an
date *.11.01	tous les premiers Novembre
}	Fin de afterwork

Clients :

`src` admins {	définit un client :
ip 172.16.12.0	adresse IP
ip 192.168.2.0-192.168.2.255	plage d'adresses.
}	

Destinations : dans cet exemple on a mail et ads

`dst` nomdest {	définit une destination :
domainlist fichier	liste de domaines
urllist fichier	liste d'urls.
log nomdest	tracer dans les logs
}	

Déclaration de l'acl :

acl {	une seule ACL peut être déclarée

// cas général
nomsource {
pass nomdest
!nomdest2
}
	La source nomsource peut aller à nomdest et non pas à nomdes2

nomsource **within**
afterwork {
pass all
}
 	
else {
pass !ads !mail all
redirect http://ser/t.htm
}
 	
	Pour nomsource on autorise tout (all est un mot prédéfinit, comme "none") si la plage horaire correspond à afterwork.
	
	Sinon (hors afterwork) on autorise tout sauf les catégories mails et ads.
	Si le client demande une page interdite il sera redirigé vers http://ser/t.htm

default {
pass none
}
	pour toutes les autres sources on n'autorise rien.

}	Fin de l'ACL

Tableau 7 : les composants du fichier de configuration de squidGuard.

III. 4. Intégration de SquidGuard dans Squid :

Cette opération est réalisée de manière extrêmement simple, il suffit d'éditer le fichier de configuration de squid (squid.conf) et de rajouter :

```
#  TAG: redirect_program
#       Specify the location of the executable for the URL redirector.
#       Since they can perform almost any function there isn't one
included.
#       See the FAQ (section 15) for information on how to write one.
#       By default, a redirector is not used.
#
#Default:
# none
# redirect_program <emplacement de SquidGuard> -c <emplacement du
# fichier de configuration>
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

III. 5. Construction de bases de données :

SquidGuard prend à chaque lancement les fichiers textes, les lit, puis génère une base de données indexée afin d'augmenter sa rapidité. Au lieu de faire ceci à chaque lancement de Squidguard (donc à **chaque** redirecteur lancé) on peut lui préciser, à l'aide de la commande :

```
squidGuard -C all
```

de générer ces bases de données une fois pour toute. Il n'a donc plus à la faire à chaque lancement. Le gain de temps est **très** important lors de fortes utilisations.

SquidGuard ne peut exploiter ces bases que s'il fonctionne avec l' UID de squid donc ces deux commandes sont indispensables :

```
//Changement du propriétaire et du groupe du répertoire de la liste noire.
chown -R squid:squid /var/squidGuard/blacklists
//  Changement des droits d'accès
chmod -R 760 /var/squidGuard/blacklists
```

III. 6. Les listes noires (Blacklists)

Les listes noires sont des ensembles de domaines et /ou d'URL qui définissent des cibles potentielles pour les internautes d'un LAN.

Au moyen de ces listes noires, on peut définir des cibles accessibles, ou interdites. Par exemple, on peut définir des ensembles de domaines ou d'URL qui seront les seuls à pouvoir être atteints par certains "source groups" ou, à l'inverse, définir des ensembles de domaines ou d'URI dont l'accès sera interdit.

- **Personnalisation de la liste noire :**

Il est parfois nécessaire de personnaliser les listes noires afin de l'adapter à notre usage. Pour cela il suffit de créer un fichier domains.diff ou urls.diff dans le répertoire concerné qui contient les urls et les domaines à ajouter (on met un + devant) ou à enlever (on met un - devant) puis de lancer la commande :

```
squidguard -u.
```

Il faudra ensuite enlever les .diff crées. Exemple d'un domains.diff.

```
-yahoo.com
+cooltext.com
+chat.com
```

Service de routage et de firewall (IPTABLES)

VI.1.Introduction

Les réseaux locaux utilisent les adresses privées, qui ne sont pas routables sur Internet. Il est donc évident que les machines du réseau local ne pourront pas envoyer de paquets sur Internet, et ne recevront jamais de paquets provenant d'Internet. Heureusement, il existe une technique nommée *masquerading* , basée sur un mécanisme de translation d'adresse (« NAT » en anglais, abréviation de « Network Address Translation »), qui permet de modifier les paquets émis à destination d'Internet à la volée afin de pouvoir partager une connexion Internet même avec des ordinateurs qui utilisent des adresses locales.

Il faut toutefois bien se rendre compte que le fait que fournir un accès à Internet à un réseau local pose des problèmes de sécurité majeurs. Pour des réseaux locaux familiaux, les risques de piratages sont bien entendu mineurs, mais lorsque la connexion à Internet est permanente ou lorsque les données circulant sur le réseau local sont sensibles, il faut tenir compte des risques d'intrusion.

Lorsqu'on utilise des adresses IP dynamiques, il est relativement difficile d'accéder à des machines du réseau local, sauf si la passerelle expose des services internes au reste du réseau. En revanche, si les adresses utilisées sont fixes et valides sur Internet, le risque devient très important. La configuration de la passerelle doit donc se faire avec soin dans tous les cas, et l'installation d'un Firewall est plus que recommandée (un « firewall », ou « pare-feu » en français, est un dispositif qui consiste à protéger un réseau local).

Linux est capable de filtrer les paquets circulant sur le réseau et d'effectuer des translations d'adresses depuis la version 2.0. Cependant, les techniques utilisées ont été profondément remaniées à chaque version. À partir de la version 2.4.0, une architecture extensible a été mise en place et semble répondre à tous les besoins de manière simple : *IPTABLES* **[WWW_4].**

VI.2.Présentation d'IPTABLES :

IPTABLES est un Firewall implémenté au niveau du noyau Linux 2.4. Il remplace le Firewall ipchains datant du noyau 2.2 et permet de faire du **firewalling à état** appelé également **Stateful**, de la translation de port et d'adresse, du filtrage au niveau port et IP et beaucoup d'autres choses comme le "Mangle" ou modification des paquets à la volée.

Note :

On parle souvent de « IPTABLES » par abus de langage, alors que le nom du logiciel est « Netfilter ».IPTABLES est en fait juste l'outil qui permet à un administrateur de configurer Netfilter en mode utilisateur (interface lignes de commande).

VI.3.Installation

Pré requis :

IPTABLES est installé en standard sur de nombreuses distributions Linux récentes. En particulier, il est installé sur Fedora core 6.

Options de compilation du kernel :

Dans le cas d'une recompilation du kernel, il faut spécifier les options nécessaires au fonctionnement d'Iptables. Les options suivantes doivent être activées en tant que module :

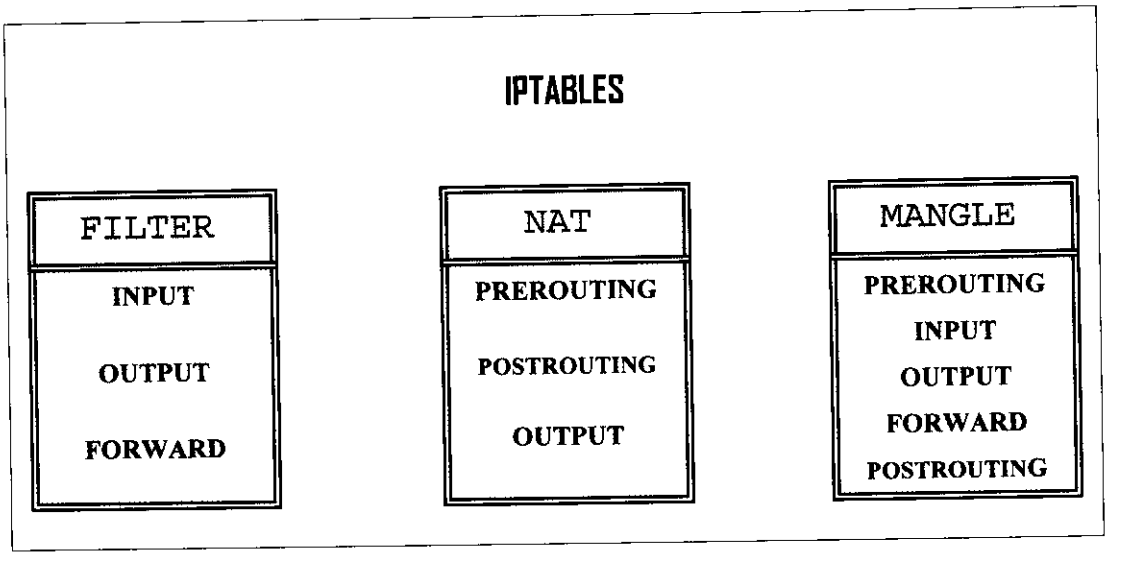
```
CONFIG_PACKET
CONFIG_NETFILTER
CONFIG_IP_NF_CONNTRACK
CONFIG_IP_NF_FTP
CONFIG_IP_NF_IRC
CONFIG_IP_NF_IPTABLES
CONFIG_IP_NF_FILTER
CONFIG_IP_NF_NAT
CONFIG_IP_NF_MATCH_STATE
CONFIG_IP_NF_TARGET_LOG
CONFIG_IP_NF_MATCH_LIMIT
CONFIG_IP_NF_TARGET_MASQUERADE
```

Et éventuellement :

`CONFIG_IP_NF_COMPAT_IPCHAINS`	pour garder la compatibilité avec ipchains.
`CONFIG_IP_NF_COMPAT_IPFWADM`	pour garder la compatibilité avec ipfwadm.
`CONFIG_IP_NF_TARGET_REDIRECT`	indispensable, pour les proxies transparents.
`CONFIG_IP_NF_MATCH_MAC`	permet de matcher avec les adresses MAC.

VI.4. Les tables :

Une table permet de définir un comportement précis d'IPTABLES. Une table est en fait un ensemble de chaînes, elles-mêmes composées de règles. IPTABLES fonctionne avec trois tables FILTER, NAT et MANGLES.



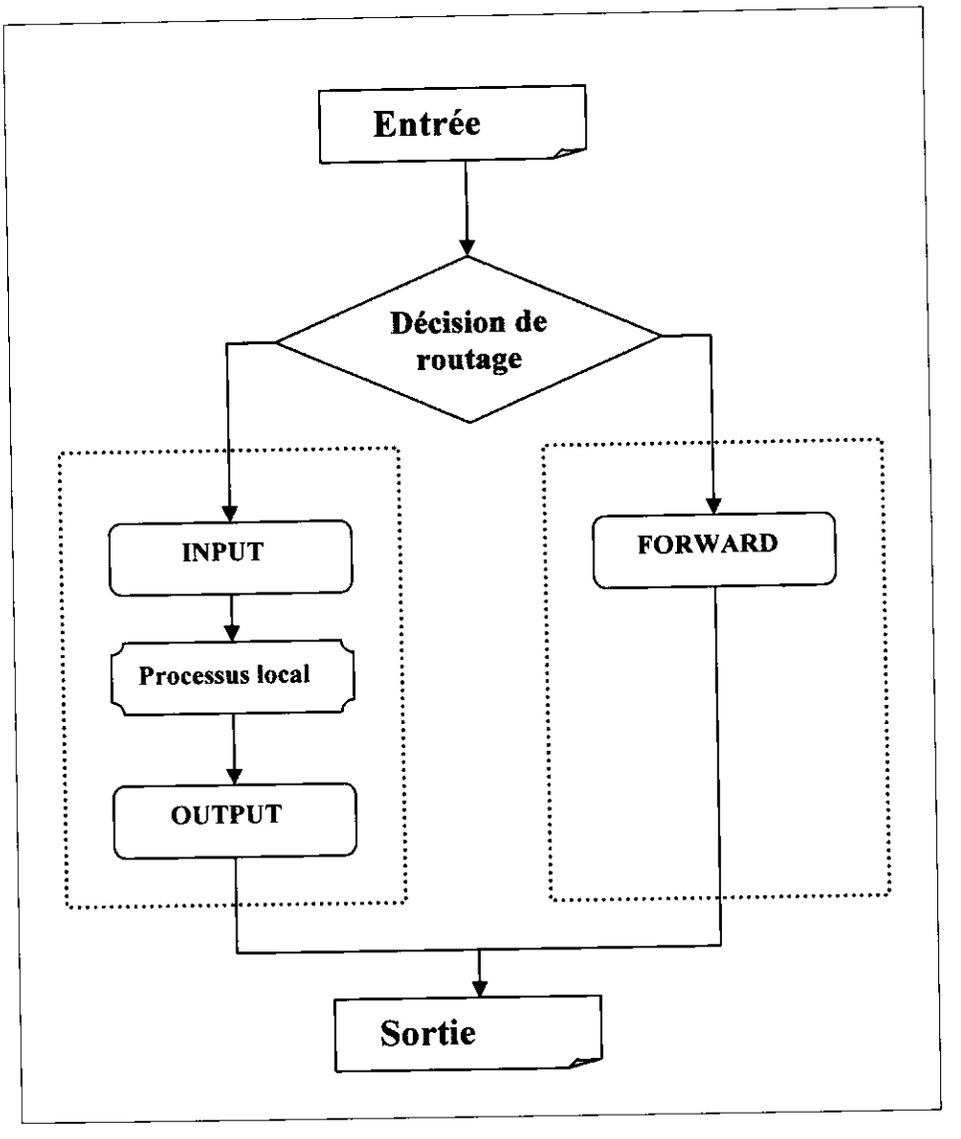
- Figure 2.1-

VI.4.1.La table FILTER:

La table FILTER est la table par défaut lorsqu'aucune table n'est spécifiée. Elle va contenir toutes les règles qui permettront de filtrer les paquets. Elle contient 3 chaînes:

- La chaîne **INPUT** filtre les paquets entrant localement sur l'hôte.
- La chaîne **OUTPUT** filtre les paquets émis par l'hôte local.
- La chaîne **FORWARD** filtre les paquets routés par l'hôte.

Déroulement



- Figure 2.2-

Un paquet entre au niveau du pare-feu. Peu importe par quelle interface il entre, il peut venir aussi bien du réseau local que d'un réseau externe (par exemple Internet). Il passe d'abord par la fonction de décision de routage. C'est elle qui va déterminer si le paquet est destiné à un processus local de l'hôte ou à un hôte sur un autre réseau.

- Si le paquet est destiné à l'hôte local:

1. Il traverse la chaîne INPUT
2. S'il n'est pas rejeté, il est transmis au processus impliqué. Ce processus va donc le traiter et éventuellement émettre un nouveau paquet en réponse.
3. Ce nouveau paquet traverse la chaîne OUTPUT
   · S'il n'est pas rejeté, il va vers la sortie.

- Si le paquet est destiné à un hôte d'un autre réseau:

1. Il traverse la chaîne FORWARD.
2. S'il n'est pas rejeté, il poursuit alors sa route.

Résumé :

Seuls les paquets destinés à un processus local traversent la chaîne INPUT.
Seuls les paquets issus d'un processus local traversent la chaîne OUTPUT.
Seuls les paquets destinés au routage traversent la chaîne FORWARD.

VI.4.2.La table NAT (Network Address Translation) :

Cette table est utilisée pour la translation d'adresse ou la translation de port. Il existe 3 chaînes :

1. La chaîne **PREROUTING**, permet de faire la translation d'adresse de destination. Cette méthode permet de faire croire au réseau externe qu'il y a un serveur WEB par exemple sur le port 80 du Firewall, alors que ce dernier est situé sur un serveur du réseau privé.
2. La chaîne **POSTROUTING**, permet de faire la translation d'adresse source. Cette méthode est utilisée pour connecter un réseau privé à Internet, en n'utilisant qu'une seule adresse IP publique.
3. La chaîne **OUTPUT** permet de modifier la destination de paquets générés par le Firewall lui-même.

La translation d'adresse appelée également NAT, permet non seulement de faire de la translation d'adresses, mais également de la translation de ports. Les fonctions NAT permettent donc de changer l'adresse IP, le port de l'émetteur ou du destinataire d'un paquet transitant par le Firewall.

Cela sert à une quasi infinité de choses comme :

- Le masquage d'adresse :

C'est une fonction fondamentale lorsque l'on souhaite connecter un réseau privé à l'Internet et qu'on ne dispose que d'une seule IP valide sur le Net. Les clients sont sur le réseau privé et les serveurs sont sur le Net.

- Le NAT de destination :

Cela permet de résoudre les problèmes qui apparaissent dans l'autre sens. Les clients sont sur le Net et les serveurs sont sur le réseau privé. Il est possible qu'ayant une seule IP valide sur le Net, nous voulions tout de même offrir des services tels que HTTP, FTP, SMTP, POP et peut-être d'autres encore.

VI.4.3.La table MANGLE :

Cette table permet le marquage des paquets. Jusqu'au noyau 2.4.17, elle offrait deux chaînes prédéfinies : **PREROUTING** (pour modifier les paquets entrants, avant le routage) et **OUTPUT** (pour modifier les paquets générés localement, avant le routage). Depuis le noyau 2.4.18, trois autres chaînes prédéfinies sont aussi prises en charge : **INPUT** (pour les paquets entrants, destinés à la machine elle-même), **FORWARD** (pour modifier les paquets routés à travers la machine) et **POSTROUTING** (pour modifier les paquets lorsqu'ils sont sur le point de sortir).

IV. 5.Les cibles :

Les cibles correspondent aux actions qu'il est possible d'effectuer lorsqu'un paquet correspond aux critères définis par l'une des règles. Il y a cinq cibles disponibles par défaut, pour toutes les tables. Toutefois, chaque table a la possibilité d'avoir des cibles qui lui sont propres, et un certain nombre de cibles développées à la base en tant que simples extensions sont désormais distribuées par défaut avec Iptables.

- ***ACCEPT*** : pour accepter le paquet qui vérifie le critère de sélection de la règle.
- ***DROP :*** pour éliminer purement et simplement le paquet.

- ***REJECT*** : pour rejeter le paquet (en signalant l'erreur à la machine émettrice). Cette cible n'est utilisable que dans les chaînes `INPUT`, `FORWARD` et `OUTPUT`, ainsi que dans les chaînes utilisateurs appelées à partir de ces chaînes.

- ***REDIRECT :*** pour rediriger le paquet sur une autre machine, souvent la machine locale. Cette cible n'est utilisable qu'avec la table `nat`, car il s'agit d'une translation d'adresse. On ne peut l'utiliser que dans les chaînes `PREROUTING` et `OUTPUT` et les chaînes utilisateur appelées à partir de ces chaînes.

- ***SNAT:*** pour permettre la modification de l'adresse source du paquet. Cette cible n'est bien entendu accessible qu'au niveau de la table `nat`. Comme la modification de l'adresse source n'a de signification que pour les paquets devant sortir de la passerelle, cette cible ne peut être utilisée que dans la chaîne `POSTROUTING` et les chaînes utilisateur appelées à partir de cette chaîne.

- ***DNAT*** : pour effectuer la modification de l'adresse destination du paquet, par exemple afin de le détourner vers une autre machine que celle vers laquelle il devait aller. Cette cible n'est accessible que dans la table `Nat`, et n'est utilisable que dans les chaînes `PREROUTING` et `OUTPUT` ainsi que dans les chaînes utilisateur appelées à partir de ces chaînes.

- ***MASQUERADE*** : pour effectuer une translation d'adresse sur ce paquet, dans le but de réaliser un partage de connexion à Internet. Cette cible n'est accessible que dans la chaîne `POSTROUTING` de la table `Nat`, ainsi que dans les chaînes utilisateur appelées à partir de cette chaîne.

IV. 6. Le suivi de connexion :

Le suivi de connexion est un concept essentiel dans IPTABLES. C'est une sorte d'intelligence artificielle qui permet d'établir des liens de cause à effet entre les paquets qui passent dans la pile. Il existe 4 états :

NEW : Les paquets sont les premiers d'une connexion et tentent de l'établir, il ne faut donc autoriser que ceux des types et destinations souhaitées.

ESTABLISHED : Les paquets proviennent d'une connexion déjà établie et donc en suivent et en précèdent d'autres, il est donc correct d'accepter ceux-ci sans trop de restriction en même temps que NEW.

RELATED : Les paquets RELATED sont ceux qui établissent une nouvelle connexion, mais qui sont en rapport avec une connexion déjà existante. L'état RELATED peut être utilisé pour filtrer des connexions qui font partie d'un protocole multi-connexion, comme FTP, ainsi que les paquets d'erreur en rapport avec des connexions existantes (comme les paquets d'erreur ICMP).

INVALID : Les paquets sont dans un format de connexion invalide, il est donc conseillé de rejeter ou détruire ceux-ci.

Ces critères permettent encore de préciser le filtrage et donc éviter énormément d'attaques.

IV. 7. Les commandes d'Iptables

Il est possible, grâce à iptables, d'effectuer toutes les opérations d'administration désirées sur les tables de Netfilter.il est donc possible de créer de nouvelles chaînes,de les détruire,de définir les politiques par défaut, ainsi que de manipuler les règles de ces chaînes (en ajouter, en supprimer ou en remplacer une). En fait, iptables dispose d'un grand nombre d'options.

IV. 7. 1. Manipulation des chaînes :

Toutes les options peuvent être utilisées avec toutes les tables gérées par le noyau. La table sur laquelle une commande s'applique peut être précisée à l'aide de l'option `-t`. Si cette option n'est pas fournie, la table utilisée par défaut sera la table `filter`, qui est celle qui est normalement utilisée pour définir des règles de filtrage des paquets dans un Firewall.

La création d'une nouvelle chaîne se fait simplement avec l'option `-N` :

```
iptables [-t table] -N chaîne
```

Où `chaîne` est le nom de la chaîne à créer. Il est interdit d'utiliser un des mots clés réservés par **iptables**. En pratique, il est recommandé d'utiliser des noms de chaînes en minuscules, car les chaînes prédéfinies sont en majuscules.

Une chaîne peut être détruite avec l'option `-X` :

```
iptables [-t table] -X chaîne
```

Une chaîne ne peut être détruite que lorsqu'elle ne contient plus de règle. De plus, il est impossible de détruire les chaînes prédéfinies d'une table.

Pour lister l'ensemble des règles d'une chaîne avec l'option `-L` :

```
iptables [-t table] -L chaîne
```

Enfin, il est possible de supprimer toutes les règles d'une chaîne avec l'option `-F` :

```
iptables [-t table] -F chaîne
```

Où `chaîne` est le nom de la chaîne dont on veut supprimer les règles.

IV. 7. 2. Manipulation des règles :

La syntaxe générale pour ajouter une règle dans une chaîne est la suivante :

```
iptables [-t table] -A chaîne [comparaison] [-j cible]
```

Où :

- `table` est la table dans laquelle se trouve la chaîne manipulée (par défaut, il s'agit de la table `filter`) ;
- `chaîne` est le nom de la chaîne à laquelle la règle doit être ajoutée ;

▪ Les comparaisons :

Les comparaisons les plus classiques sont :

```
-p : compare au niveau protocole.
-s : compare la source des paquets.
-d : compare la destination des paquets.
-i : compare l'interface réseau d'où viennent les paquets.
-o : compare l'interface réseau par laquelle veulent sortir les paquets.
-sport  : compare le port d'où viennent les paquets.
-dport  : compare le port par lequel veulent sortir les paquets.
--state : compare l'état des connections des paquets.
```

▪ Suppression d'une chaîne :

La suppression d'une règle d'une chaîne se fait avec la commande suivante :

```
iptables -D chaîne numéro
```

Où `chaîne` est la chaîne dont on veut supprimer la règle, et `numéro` est le numéro de la règle à supprimer dans cette chaîne. Il est également possible d'utiliser l'option `-D` avec les mêmes options que celles qui ont été utilisées lors de l'ajout de la règle, si l'on trouve cela plus pratique.

IV. 7. 3. Politique par défaut:

Enfin, la politique d'une chaîne, c'est-à-dire la cible par défaut, peut être fixée avec l'option « -P » :

```
iptables -P chaîne cible
```

Où chaîne est l'une des chaînes prédéfinie, et cible est l'une des cibles ACCEPT ou DROP. Remarquez que l'on ne peut pas définir de politique pour les chaînes créées par l'utilisateur, puisque les paquets sont vérifiés avec les règles restantes de la chaîne appelante lorsqu'ils sortent de la chaîne appelée.

IV. 7. 4. Spécificités NAT :

--to-destination : Utilisé en destination pour le DNAT, permet de spécifier l'adresse de destination de la translation, on peut également spécifier un port s'il est différent du port source. **--to-source :** Utilisé pour le SNAT, permet de spécifier l'adresse source de la translation.

to-ports : précise le port de la destination.

IV. 7. 5. Exemple de quelques commandes d'iptables :

Afin de mieux comprendre voici quelques exemples de commandes.

Pour la table NAT :

- *Ajout d'une règle de DNAT :*

```
# iptables -t nat -A PREROUTING --dst 'adresse-Internet-du-firewall' -p tcp
--dport 'port' -j DNAT \ --to-destination 'adresse-de-destination'
```

- *Ajout d'une règle de SNAT :*

iptables -t nat -A POSTROUTING -p tcp --dst 'adresse-locale' --dport 'port' -j SNAT \ --to-source 'adresse-Internet-du-firewall'

Pour la table FILTER :

Suppression des paquets venant de 10.0.0.0 :

```
# iptables -P INPUT -s 10.0.0.0 DROP
```

Autorisation de la boucle locale 127.0.0.1 :

```
# iptables -P INPUT -s 127.0.0.1 ACCEPT
```

IV. 8. Proxy transparent

Pour utiliser un Proxy, l'utilisateur doit paramétrer son navigateur pour qu'il envoie ses requêtes sur le port d'écoute du Proxy. Il y a donc une configuration à faire au niveau du navigateur.

Il est possible de rendre le Proxy transparent, c'est-à-dire donner l'illusion à l'utilisateur, qu'il interroge directement le serveur distant. Pour ce faire, on redirige les requêtes du navigateur vers le port d'écoute de Squid. Cette redirection est possible grâce à iptables.

Il faudra ajouter dans /etc/squid/squid.conf

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Activer le routage d'adresses par la commande :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Puis ajouter la règle Iptables pour la redirection des requêtes sur le port 80

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
                                          -j REDIRECT --to-port 3128
```

Les clients peuvent envoyer leurs requêtes sur le port 80 du proxy le service NAT du routeur les redirige sur le port 3128.

Serveur de messagerie (Postfix)

V.I- Introduction

Un serveur de messagerie électronique est un logiciel serveur de courrier électronique. Il a pour vocation de transférer les messages électroniques d'un serveur à un autre. Un utilisateur n'est jamais en contact direct avec ce serveur mais utilise soit un client de messagerie, soit un courrielleur web, qui se charge de contacter le serveur pour envoyer ou recevoir les messages.

La plus part des serveurs des messageries possèdent ces deux fonctions (envoi/réception), mais elles sont indépendantes et peuvent être dissociées physiquement en utilisant plusieurs serveurs **[WWW_5]**.

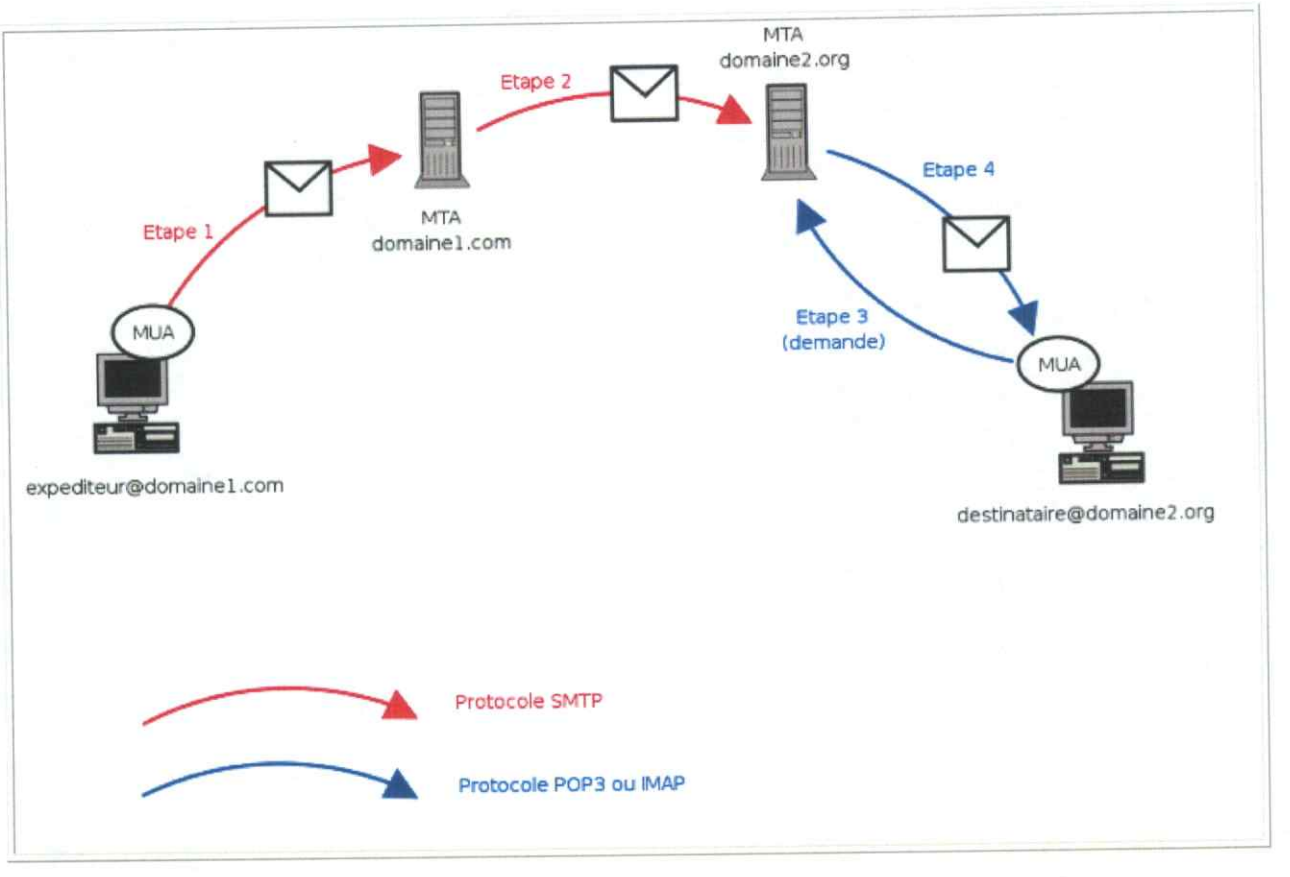
Envoi

Entre l'utilisateur et son serveur, l'envoi d'un courrier électronique se déroule généralement via le protocole SMTP. Puis c'est au serveur d'envoyer le message au serveur du destinataire, cette fonction est appelée Mail Tranfer Agent en anglais, ou MTA.

Réception

La réception d'un courrier électronique s'effectue elle aussi en deux temps. Le serveur doit recevoir le message du serveur de l'expéditeur, il doit donc gérer des problèmes comme un disque plein ou bien une corruption de la boite aux lettres et signaler au serveur expéditeur toute erreur dans la délivrance. Il communique avec ce dernier par l'intermédiaire des canaux d'entrée-sortie standard ou bien par protocole spécialisé comme LMTP (Local Mail Tranfer Protocol).Cette fonction de réception est appelée Mail Delivery Agent en anglais, ou MDA. Le serveur doit renvoyer le message au destinataire final lorsque celui ci le désire, généralement via le protocole POP3 ou IMAP.

Voici quelques serveurs parmi les plus connus : Qmail, Sendmail, Postfix, Exim, Microsoft Exchange Server.



- Figure 2.3-

V.II- Présentation de Postfix :

Postfix est le système de courrier crée par Wietse Venema, également auteur des TCP wrappers, reconnus pour leur intérêt dans le domaine de la sécurité, et pour la qualité du code par toute la communauté des logiciels libres.
Nommé Secure Mailer lors de la première version en 1998, il fut publié comme logiciel libre (IBM public licence) sous le nom de Postfix.

Avec près de 100 millions d'utilisateurs, ce sont des milliards de mails qui circulent chaque jour. L'objectif de la conception de Postfix était de réaliser un système de courrier alternatif à Sendmail (environ 70% des serveurs de messagerie), qui soit rapide, facile à administrer et sécurisé tout en étant autant que possible compatible avec Sendmail.

V.III- Objectifs principaux de Postfix :

1) Large diffusion : Postfix doit être largement diffusé afin d'avoir un impact sensible sur les performances et la sécurité des systèmes de messagerie sur l'Internet. C'est pourquoi Postfix est un logiciel libre sans aucune restriction.

2) Performance : Postfix est au moins trois fois plus rapide que son proche « rival » (Qmail de D.Berbstein).Un serveur Postfix sur un PC de bureau peut envoyer et recevoir quotidiennement un million de mails différents.
Postfix utilise les « trucs » utilisés sur les serveurs Web pour réduire la création de processus, et d'autres types de « trucs » permettent de réduire l'utilisation du système de fichiers, tout ça sans compromettre les performances.

3) Compatibilité : Postfix a été conçu pour être compatible avec Sendmail afin de faciliter la migration. Les fichiers /var[/spool]/mail, /etc/aliases, NIS, et ~/.forward sont utilisés.
Toutefois, l'administration devant être simple, Postfix n'utilise pas de fichier sendmail.cf.

4) Sûreté et robustesse : Postfix se comporte de façon rationnelle face au nombre de tâches demandées. Si le système n'a plus de mémoire ou d'espace disque, Postfix n'endommagera pas les choses d'avantage, il a été conçu pour rester sous contrôle.

5) Flexibilité : Postfix est conçu de façon modulaire, une douzaine de petits programmes effectuent des tâches bien précises.
Il est possible de remplacer les programmes par défaut par des produits maison, voire de supprimer certains programmes inutiles dans certains cas (un firewall ou une station de travail n'a pas besoin de livrer localement des mails).

6) Sécurité : Postfix utilise plusieurs niveaux de défense afin de protéger le système de toute intrusion. Chaque programme est enfermé dans sa cage (chrooté), il n'y a aucun lien direct entre le réseau et les programmes sensibles comme la livraison du courrier local. Postfix ne fait même pas confiance à ses propres files de courier. De plus, aucun des programmes n'est SUID.

V.4- Commandes :

Nous parlerons dans cette partie, des utilitaires "ligne de commandes" livrées avec Postfix qui servent à maintenir le système.

Le système Postfix possède une collection de commandes très utiles, pour des raisons d'homogénéité, elles seront toutes nommées postquelquechose.

- La commande **postfix** contrôle le système de courrier.
  C'est l'interface pour démarrer et arrêter le système, et pour quelques autres opérations administratives.
  Cette commande est réservée au super utilisateur.
  Exemples :
  #/etc/init.d/postfix restart force postfix à relire ses fichiers de configuration (après modification du /etc/postfix/main.cf).
  #/etc/inid.d/postfix check vérifie la configuration du système de courrier.
  #/etc/inid.d/postfix flush force postfix à tenter de vider la file deferred, donc à envoyer les messages en attente.
- La commande **postalias** sert à convertir le fichier aliases en format bases de données (*.db).
  Ce programme se cache derrière la commande newaliases.
- La commande **postcat** montre les contenus de la file d'attente de Postfix.
  C'est un programme limité, il peut être remplacé par un autre plus puissant qui permettrait d'éditer les fichiers de file d'attente de Postfix.
- La commande **postconf** montre les paramètres donnés dans le fichier main.cf de Postfix : les valeurs réelles, les valeurs par défaut, ou les paramètres qui n'ont pas de valeur par défaut.
  C'est un programme limité et primaire. Ce programme peut être remplacé par un autre plus puissant qui pourrait non seulement énumérer mais également éditer le fichier main.cf.

Exemples :

#postconf -n affiche les paramètres modifiés par notre configuration.

#postconf -d affiche les paramètres par défaut.

- La commande **postmap** sert à convertir en format base de données des tables de consultation de Postfix telles que canonical et d'autres.
  C'est un cousin de la commande de makemap d'UNIX.
- La commande **postqueue** est l'utilitaire lancée par la commande de sendmail pour vider ou lister la file d'attente du courrier.
- La commande **postsuper** sert à la maintenance de la file d'attente de Postfix.
  Cette commande est lancée lors du démarrage du système de courrier.

V.5- Configuration de Postfix :

Postfix a plusieurs paramètres de configuration qui sont contrôlés par l'intermédiaire du fichier **main.cf**. Heureusement, ils ont des valeurs par défaut. Dans la plupart des cas, pour pouvoir utiliser le système de courrier Postfix il suffit de renseigner deux ou trois paramètres :

- Le nom de domaine utilisé pour le courrier sortant.
- Les domaines terminaux pour l'acheminement du courrier.
- Les clients autorisés à utiliser Postfix.

Ces valeurs seront utilisées pour définir les valeurs par défaut de nombreux autres paramètres.

V.5.1-Fichier de configuration de Postfix :

Par défaut, les fichiers de configuration de Postfix se trouvent dans le répertoire **/etc/postfix**. Les deux plus importants sont **main.cf** et **master.cf**, ces fichiers doivent appartenir à root. Donner à quelqu'un d'autre les droits d'écriture sur ces deux fichiers (ou sur leurs répertoires parents) revient à lui donner des privilèges root.
Un minimum de paramètres doit être configuré dans **/etc/postfix/main.cf**. Les paramètres ressemblent à des variables shell avec la différence importante : c'est que Postfix ne sait pas interpréter les apostrophes comme un shell Unix.

- Pour renseigner un paramètre :
  /etc/postfix/main.cf:

```
parameter = value
```

- Et pour l'utiliser, il suffit de le faire précéder par un $:

/etc/postfix/main.cf:

```
other_parameter = $parameter
```

A chaque changement des fichiers **main.cf** ou **master.cf**, il faut lancer la commande suivante en tant que root pour prendre en considération ces changements : **# postfix restart**.

V.5.2- Paramètres de configuration de Postfix :

Les paramètres les plus importants à configurer sont décrits ci-dessus :

- Le nom du serveur Postfix :

Le paramètre **myhostname** décrit le nom complet et conforme (fqdn) de la machine utilisant le système Postfix.

$myhostname est la valeur par défaut ainsi que dans beaucoup d'autres paramètres de configuration de Postfix.

- Le domaine de Postfix :

Le paramètre **mydomain** indique le domaine parent de **$myhostname**.

Par défaut, il vaut **$myhostname** sans la première partie du nom.

- Le nom de domaine utilisé pour le courrier sortant :

Le paramètre **myorigin** indique le "vrai" nom du serveur qui apparaîtra dans tout courrier posté sur cette machine.

La valeur par défaut est le nom local de machine, **$myhostname**, qui a pour valeur par défaut le nom de la machine.

A moins de ne travailler que sur un très petit domaine, la valeur du paramètre **$mydomain**, qui a pour valeur par défaut le nom du domaine de la machine, peut être préférable.

Ce paramètre permet de renseigner postfix sur la machine qui a posté.

Exemple :

```
myorigin = $myhostname (défaut)
myorigin = $mydomain (peut être choisi)
```

- Les domaines terminaux pour l'acheminement du courrier :

Le paramètre **mydestination** indique à Postfix les domaines pour lesquels cette machine délivrera le courrier localement.

Exemples :

- Par défaut :

```
mydestination = $myhostname localhost.$mydomain
```

- Serveur de messagerie pour tout un domaine :

```
mydestination = $myhostname localhost.$mydomain $mydomain
```

- Les clients autorisés à utiliser Postfix :

Par défaut, Postfix relaie le courrier des clients des réseaux autorisés et des domaines autorisés. Les réseaux autorisés sont définis par le paramètre **mynetworks**. La valeur par défaut autorise tous les clients des sous réseaux IP auxquels la machine est reliée.

Exemples :

- Par défaut :

```
mynetworks_style = subnet
```

- Sécuriser, n'autorise que la machine locale :

```
mynetworks_style = host
```

- Autoriser toutes les machines de la même classe d'adresse ip (A/B/C) que la machine utilisant Postfix, il ne faut pas utiliser pas cette option sur une connexion par modem : cela revient à autoriser tout le réseau du fournisseur d'accès.

```
mynetworks_style = class
```

- Sécuriser, n'autorise que la machine locale :

```
mynetworks =127.0.0.1/8
```

- Autoriser la machine locale et la machine dont son adresse ip est : **168.100.189.2**

```
mynetworks = 127.0.0.1/8, 168.100.189.2/32
```

Si le paramètre **mynetworks** est renseigné, Postfix ignore le paramètre **mynetworks_style.**

- Les destinations pour les quelles relier les courriers étrangers :

Par défaut, Postfix relaie le courrier étranger (provenant de clients hors réseaux autorisés) seulement vers les destinations autorisées. Les destinations extérieures autorisées sont définies avec le paramètre de configuration **relay_domains.** Par défaut, Postfix autorise tous les domaines (et sous domaines) listés dans le paramètre mydestination.

Exemples :

- Par défaut :

```
relay_domains = $mydestination
```

- Sécuriser : ne relaie aucun courrier étranger :

```
relay_domains =
```

- Relier le courrier vers le domaine de la machine utilisant Postfix, et ses sous domaines :

```
relay_domains = $mydomain
```

- Relier le courrier vers tous les domaines :

```
relay_domains = permit_all
```

- Adresse réseau proxy/Nat :

Certains serveurs sont connectés à Internet via un traducteur d'adresse (NAT) ou proxy. Cela signifie que les clients Internet se connectent sur l'adresse du traducteur ou proxy au lieu de se connecter sur l'adresse du serveur de courrier. Le traducteur ou proxy transfère la connexion sur l'adresse réseau du serveur de courrier, mais Postfix ne le sait pas.

Si Postfix est utilisé derrière un traducteur ou proxy, il faut renseigner le paramètre **proxy_interfaces** en lui indiquant toutes les adresses externes des traducteurs ou proxies. Il est possible d'utiliser des noms de machines au lieu d'adresses réseaux.

Exemple :

- Une machine derrière un traducteur :

```
proxy_interfaces = 1.2.3.4
```

V.6- Filtrage des en-têtes :

Le paramètre **header_checks** restreint les données autorisées suivant l'entête des messages reçus.

La valeur du paramètre doit pointer vers une ou plusieurs tables décrivant des en-têtes interdits.

Par défaut On permet n'importe quoi dans des en-têtes de message.

Syntaxe Indiquez une liste de tables de consultation. Toutes les fois qu'une en-tête est listée dans une table, l'action dépend du résultat de consultation:

- **REJECT :** Rejette le message, et note l'en-tête.
- **REJECT le texte ... :** Comme précédemment et envoie également le texte au créateur.
- **IGNORE :** Supprime l'en-tête du message.
- **WARN :** Note (mais ne rejette pas) l'en-tête avec un avertissement.

Forme d'une règle de filtrage : /Expression régulière/ ACTION

Exemples :

Dans le fichier **/etc/postfix/main.cf** :

```
header_checks = regexp:/etc/postfix/header_checks
header_checks = pcre:/etc/postfix/header_checks
```

Et dans le fichier **/etc/postfix/header_checks** :

- Rejeter les courriers qui portent dans leurs entêtes **Subject** (sujet) le mot publicité :

```
/^ Subject: .* publicité .* /REJECT
```

- Rejeter les courriers provenant de toto@yahoo.com

```
/^ From: toto@yahoo.com /REJECT
```

Signification des caractères spéciaux utilisés dans les expressions régulières :

Symbole	Description
^	Indique le début de la chaîne.
.	Indique n'importe quel caractère.
*	Indique 0,1 ou plusieurs occurrences du caractère de la classe précédente.

Tableau 8 : Signification des caractères spéciaux utilisés dans les expressions régulières

V.7- Réécriture des adresses des expéditeurs (correspondances canoniques) :

`Postfix` permet de modifier le champ **`From:`** qui contient l'adresse de l'expéditeur. Par défaut, cette option n'est pas active et il va donc falloir modifier la configuration pour l'utiliser.

Le principe

- indiquer à `Postfix` qu'il doit utiliser une table de réécriture des adresses des expéditeurs. Cette table s'appelle `canonical` et sera placée dans **`/etc/postfix/`**.
- Indiquer dans la table canonicale les adresses des expéditeurs à réécrire.

Exemple :

```
Nom-prénom@yahoo.com prénom
```

- A partir du fichier canonical, générer un fichier DB avec la commande : **postmap /etc/postfix/canonical.**
- Dans le fichier main.cf de postfix, il faut éditer le paramètre **sender_canonical_maps** comme suit :

```
sender_canonical_maps = hash:/etc/postfix/canonical
```

- Forcer `postfix` à relire ce fichier en faisant **postfix restart.**

V.8) Base d'alias :

La table des alias (**aliases** en anglais) fournit un large mécanisme système de redirection de mails pour les comptes locaux, et elle peut être utilisée pour créer des listes de diffusion. La redirection est assurée par un démon de livraison des mails de Postfix appelé local. Normalement, la table des alias est constituée d'un fichier texte qui sert d'entrée à la

commande **postalias** Le résultat, un fichier indexé au format **dbm** ou **db**, est utilisé pour des consultations rapides par le système de mail.

Le principe :

- Dans le fichier main.cf de postfix, il faut indiquer la table d'alias dans le paramètre **alia_maps**, et c'est fait par défaut :

```
alias_maps = hash:/etc/aliases
```

- Editer le fichier **/etc/aliases**, comme suit :

```
nom1@domaine1: nom2@domaine, nom3@domaine, nom4@domaine
```

Explication : l'alias ci-dessus veut dire que chaque courrier envoyé à l'adresse nom1@domaine1, va être redirigé à nom2@domain, nom3@domain, et nom4@domain, c'est une sorte de liste de diffusion.

- Construire la base de donnée des alias avec la commande :

postalias hash:/etc/mail/aliases

- relancer le serveur avec les commandes : **postfix restart.**

V.9-La lutte contre le spam :

Les messages non sollicités (spam) constituent actuellement l'un des problèmes les plus sérieux auxquels sont confrontés les internautes. Rien de plus énervant que de découvrir sa messagerie pleine de publicités indésirables. Mais il y a encore plus insupportable: perdre l'heure qui suit à supprimer ces publicités pour des médicaments, des solutions de refinancement et autres que les destinataires n'ont pas demandés et dont ils n'ont pas besoin. Heureusement, il existe des remèdes pour éviter ce problème. Nous avons choisi de présenter SpamAssassin.

V.9.1) Présentation de SpamAssassin :

SpamAssassin est un projet mené par la « **Apache Software Foundation** », auteur du très célèbre serveur Web **Apache HTTP Server**.

Objectif :

Le but de ce logiciel est de filtrer le trafic des courriels pour éradiquer les courriels reconnus comme Spam ou courriel non sollicité.

Face à l'augmentation importante du spam, ce logiciel connaît un engouement important et est adaptable sur de nombreux serveur de courriel dont Postfix, Sendmail, Qmail. Il peut être installé sur la plupart des systèmes basés sur Linux, Windows et Mac OS X.
SpamAssassin est distribué gratuitement sous la licence Apache Software License.

Fonctionnement :

SpamAssassin fonctionne en attribuant des "scores" à chaque message électronique selon différents tests destinés à déterminer s'il s'agit ou non d'un Spam. De nombreux tests sont proposés, pour vérifier notamment la validité des adresses de l'expéditeur et du destinataire, la validité de la date des messages, la présence dans le corps du message d'un mot répertorié sur une liste de mots interdits, l'appartenance ou non d'un des serveurs expéditeurs à une liste noire, etc. Chaque test vient grossir le score total d'un message, ceux qui dépassent un certain seuil défini par l'utilisateur sont considérés comme du spam et peuvent être soit rejetés dans la corbeille, soit arrivés à la boite de messagerie mais le sujet du message est tagué avec une mention " ***** SPAM détecté ***** ".

V.9.2) Configuration de SpamAssassin :

Pour configurer SpamAssassin, il faut éditer le fichier **/etc/mail/spamassassin/local.cf**. Par défaut, un certain nombre d'options sont prédéfinies.

- Renseigner le paramètre **required_hits** qui définit le score au delà duquel les mails sont considérés comme du spam. Plus la valeur est élevée, plus SpamAssassin laisse passer de messages non sollicités, lorsque la valeur est plus basse, le filtrage est plus strict mais présente un plus grand risque que des messages valides soient désignés à tort comme du spam.

```
required_hits  5
```

- Indiquer le texte à rajouter devant l'intitulé du message détecté comme du spam.

```
rewrite_header Subject [SPAM]
```

- Ajouter les adresses mail de confiance dans les white_list, qui permettent de ne pas considérer les courriers comme du spam.

  Exemple:

```
whitelist_from  bonne_personne@yahoo.com
```

- Ajouter les adresses mail indésirable dans les black_list, qui permettent de considérer les courriers comme du spam.

Exemple:

```
blacklist_from  mauvaise_personne@yahoo.com
```

Chapitre 3:

Conception du système

Chapitre 3

Conception du système

Introduction :

Modéliser un système avant sa réalisation permet de mieux comprendre son fonctionnement, c'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence.

Dans ce chapitre nous abordons la conception de notre système et pour la mener à bien nous avons choisi le langage de modélisation UML qui s'articule autour de neuf diagrammes selon deux modes : l'un concernant sa structure et l'autre sa dynamique de fonctionnement.

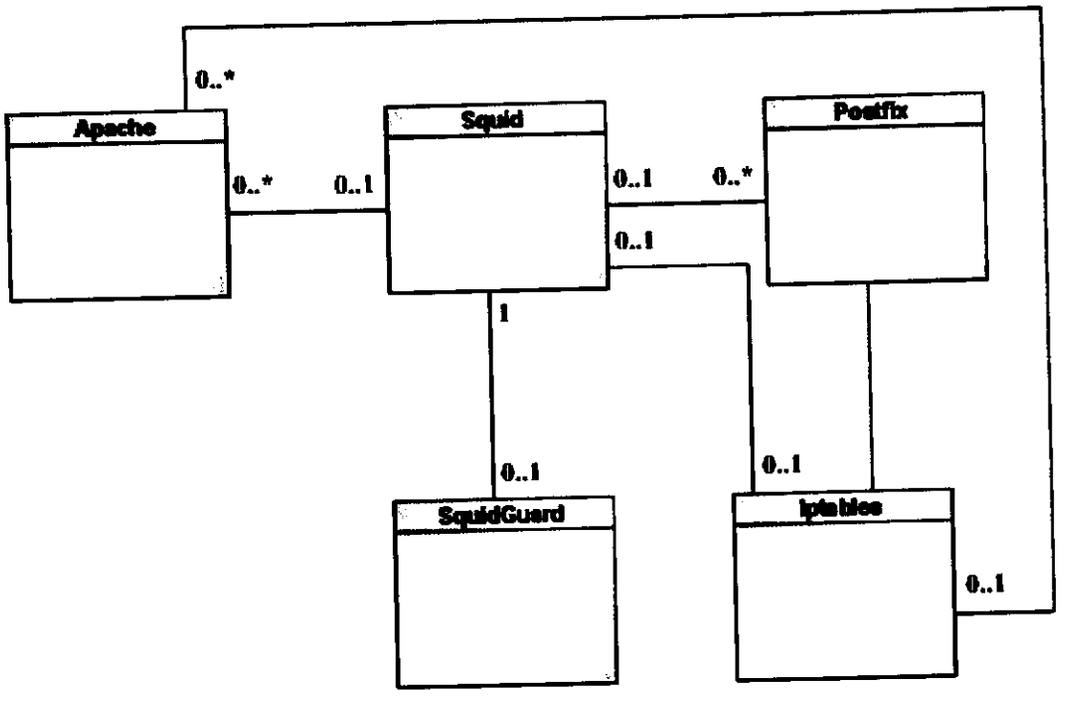
Nous avons utilisé quatre diagrammes des neufs:

- **L**es cas d'utilisation (use case), pour définir les besoins de notre système (voir le chapitre I).
- **L**e diagramme de classes, pour représenter son architecture (la vue statique).
- **L**e diagramme de séquence, pour représenter les interactions entre les objets (la vue dynamique).
- **L**e diagramme d'activité, pour modéliser le processus interactif du système.

I. Diagramme de classe:

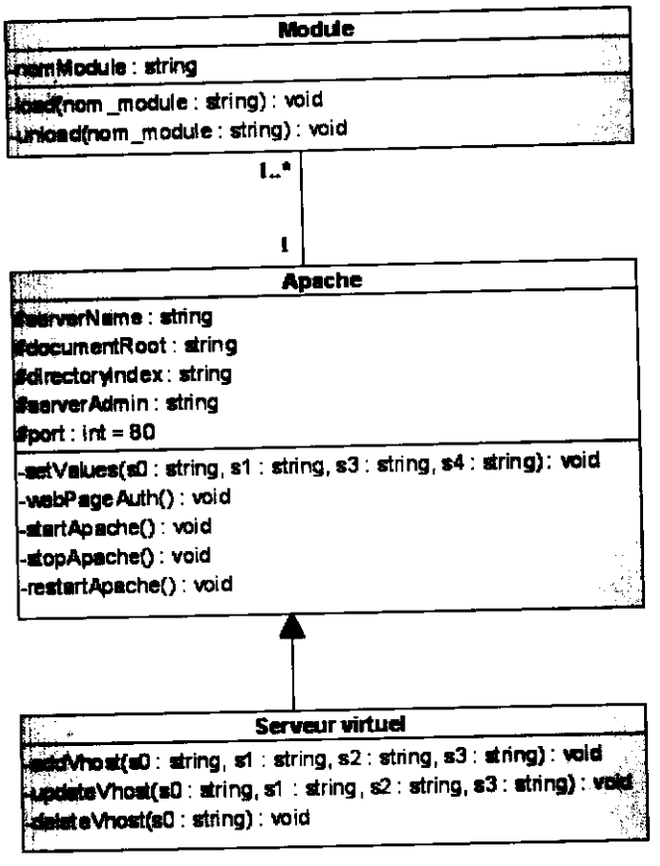
Le diagramme de classe nous permet de modéliser, d'une manière statique, les relations qui existent entre l'ensemble de classes. Il développe d'une part la structure des entités du système et d'autre part celle d'un code orienté objet **[L _1]**.

Afin de permettre au lecteur de bien comprendre notre système, nous présentons dans ce qui suit une vue globale sur l'architecture du système **(Figure 3.1)** ensuite nous allons détailler chaque module à part.



-Figure 3.1-

I.1.Diagramme de classe du serveur apache :



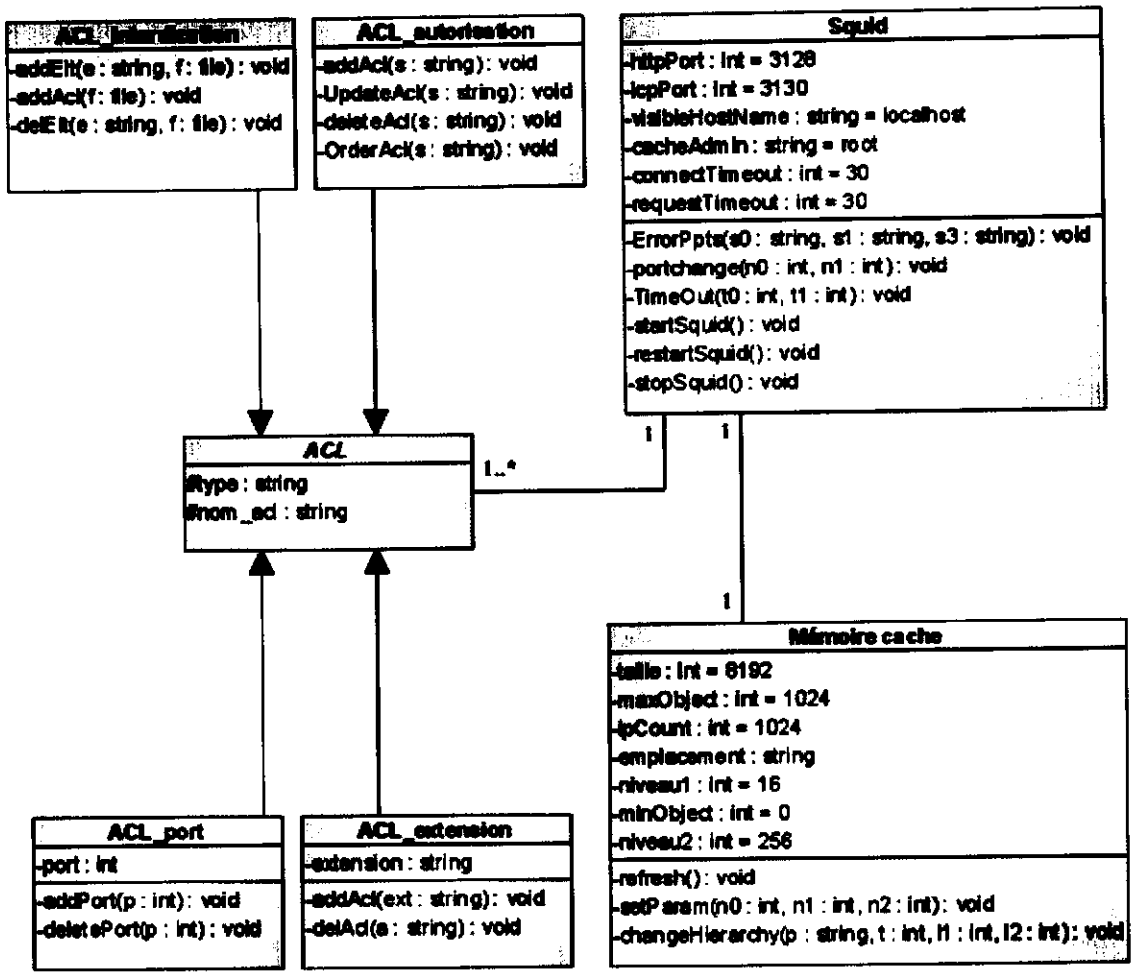
-Figure 3.2-

La classe apache représente le « serveur web », caractérisé par son nom, répertoire racine, page d'accueil, email de l'administrateur et port d'écoute.

Un serveur apache peut avoir un à plusieurs modules, chaque module est caractérisé par son nom (nomModule) et on peut le charger ou décharger.

Le serveur virtuel hérite de la classe apache, il a donc toutes les caractéristiques du serveur réel, mais les méthodes à appliquer ne sont pas les même. Pour un serveur virtuel, il est possible d'ajouter un nouveau serveur, modifier ou supprimer un serveur existant.

I.2.Diagramme de classe du serveur Squid :



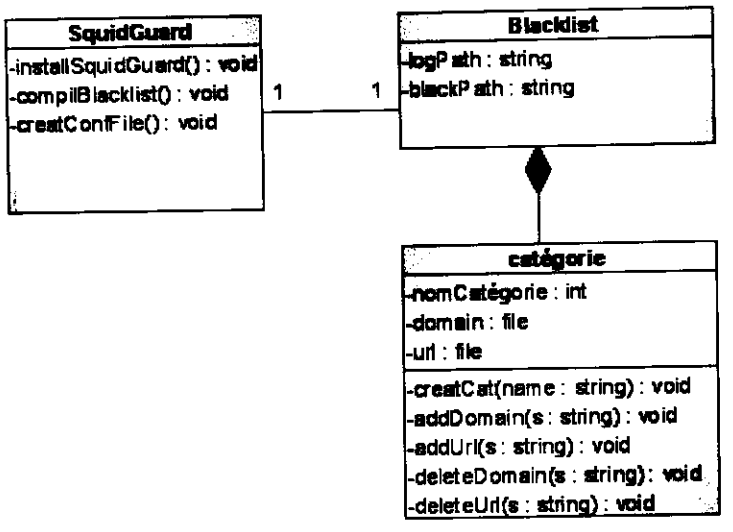
-Figure 3.3-

La classe Squid représente le proxy, qui a une seule et une seule mémoire cache. Cette dernière est représentée par son emplacement, sa taille, la taille minimale et maximale des objets stockés, le nombre d'IP stockées et le nombre de répertoire de niveau1 et niveau2.

Squid doit avoir au minimum un acl pour fonctionner comme il peut avoir plusieurs, une acl est caractérisé par son nom et son type, et qui peut être d'une des quatre catégories :

ACL_port, ACL_extension, ACL_port, ACL_autorisation et ACL_interdiction

I.3.Diagramme de classe du filtre de proxy SquidGuard :



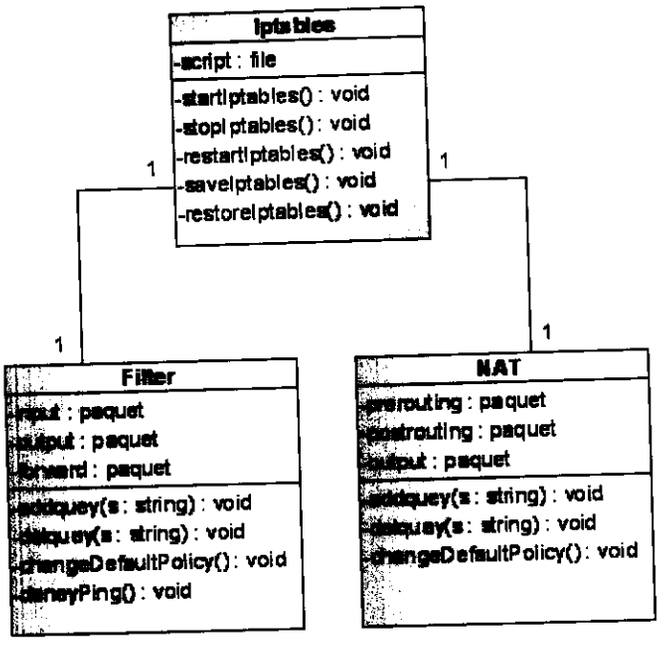
-Figure 3.4-

Il est possible d'installer squidGuard, compiler sa blacklist et créer son fichier de configuration.

SquidGuard a une et une seule blacklist qui est composé de plusieurs catégories, chaque catégorie contient un fichier domain et un fichier url où sont stockés les domaines et les urls interdits.

Cette blacklist peut être maintenue (ajout d'une nouvelle catégorie, ajout d'un domaine, ajout d'une url, suppression d'un domaine ou d'une url).

I.4.Diagramme de classe du service Iptables :

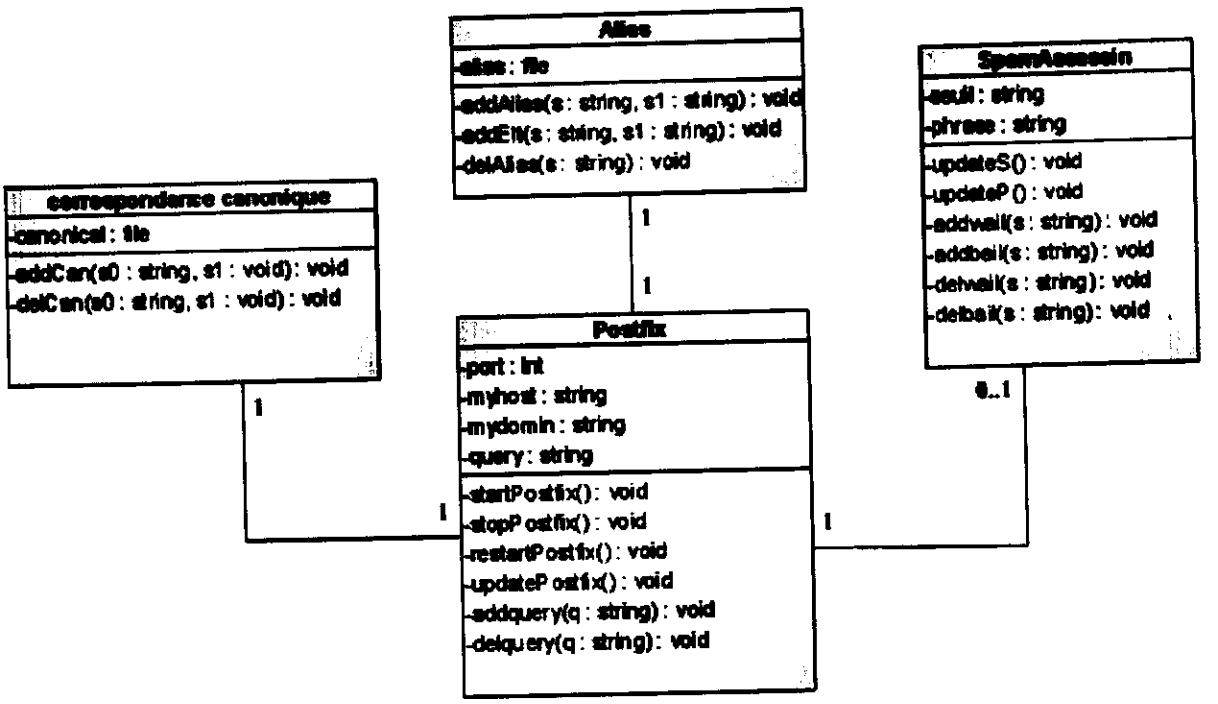


-Figure 3.5-

La classe iptables est caractérisée par son fichier script, elle a une et une seule classe Nat pour la translation d'adresses et de ports. Et de même pour la classe filter qui permet d'effectuer le filtrage des paquets.

Il est possible d'ajouter une règle de filtrage (respectivement nating), d'en supprimer une et de changer la politique par défaut.

I.5.Diagramme de classe du serveur Postfix :



-Figure 3.6-

La classe Postfix représente le serveur mail, qui est identifié principalement par le nom de la machine, le nom de domaine parent, le port, et les règles de filtrages. Postfix offre à l'utilisateur des méthodes qui lui permettent de modifier les valeurs des paramètres de Postfix, d'ajouter/ supprimer des règles de filtrage.

Postfix est en relation 1-1 avec la classe correspondance canonique, cette dernière est identifiée par le nom de la table canonical, elle offre à l'utilisateur des méthodes pour ajouter/supprimer des correspondances canoniques

La classe SpamAssassin peut être intégré dans Postfix, cette classe identifie le seuil minimal au-delà du quel un mail est considéré comme du spam, ainsi que la phrase à ajouter devant l'intitulé du mail détecté comme spam. L'utilisateur peut ajouter ou supprimer des emails dans les whitlist et les blacklist.

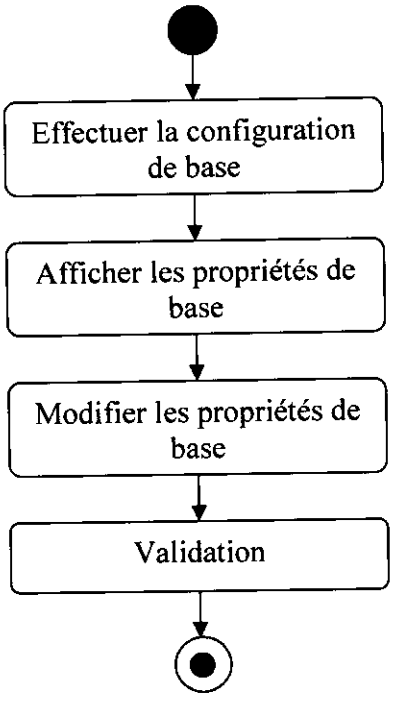
Postfix est en relation 1-1 avec la classe Alias qui se charge de la tâche de redirection des emails. Dans cette table on peut ajouter ou supprimer des alias.

II. Le diagramme d'activité

Le diagramme d'activité est principalement un organigramme qui montre le flot de contrôle d'une activité à l'autre, utilisé pour modéliser les aspects dynamiques d'un système. Cela implique la modélisation des étapes séquentielles (voire concurrentes) dans un processus de calcul **[L _1]**.

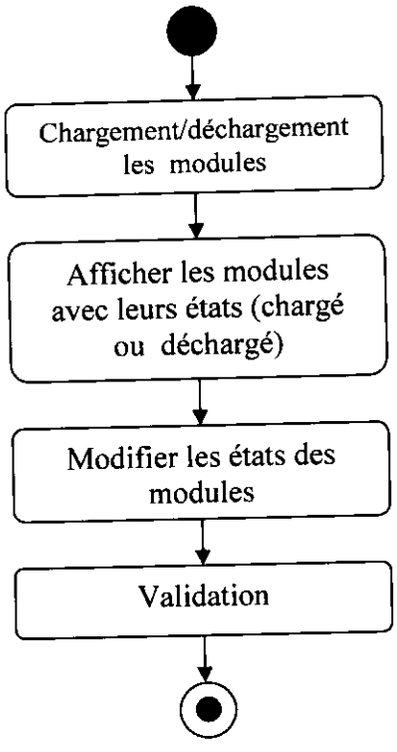
Nous présentons dans ce qui suit les principaux diagrammes d'activités de notre système.

1_Diagramme d'activité pour la configuration de base d'Apache :



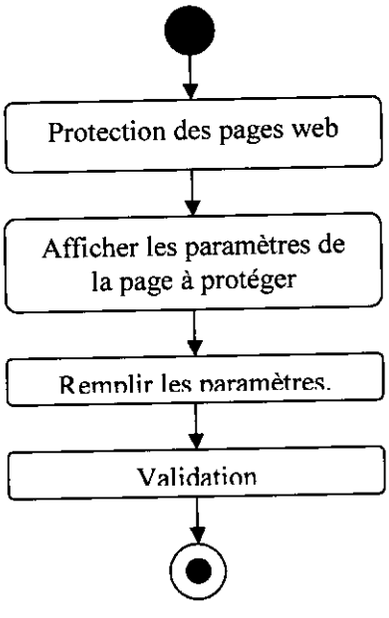
-Figure 3.8-

2_**Di**agramme d'activité pour le chargement/ déchargement des module d'Apache :



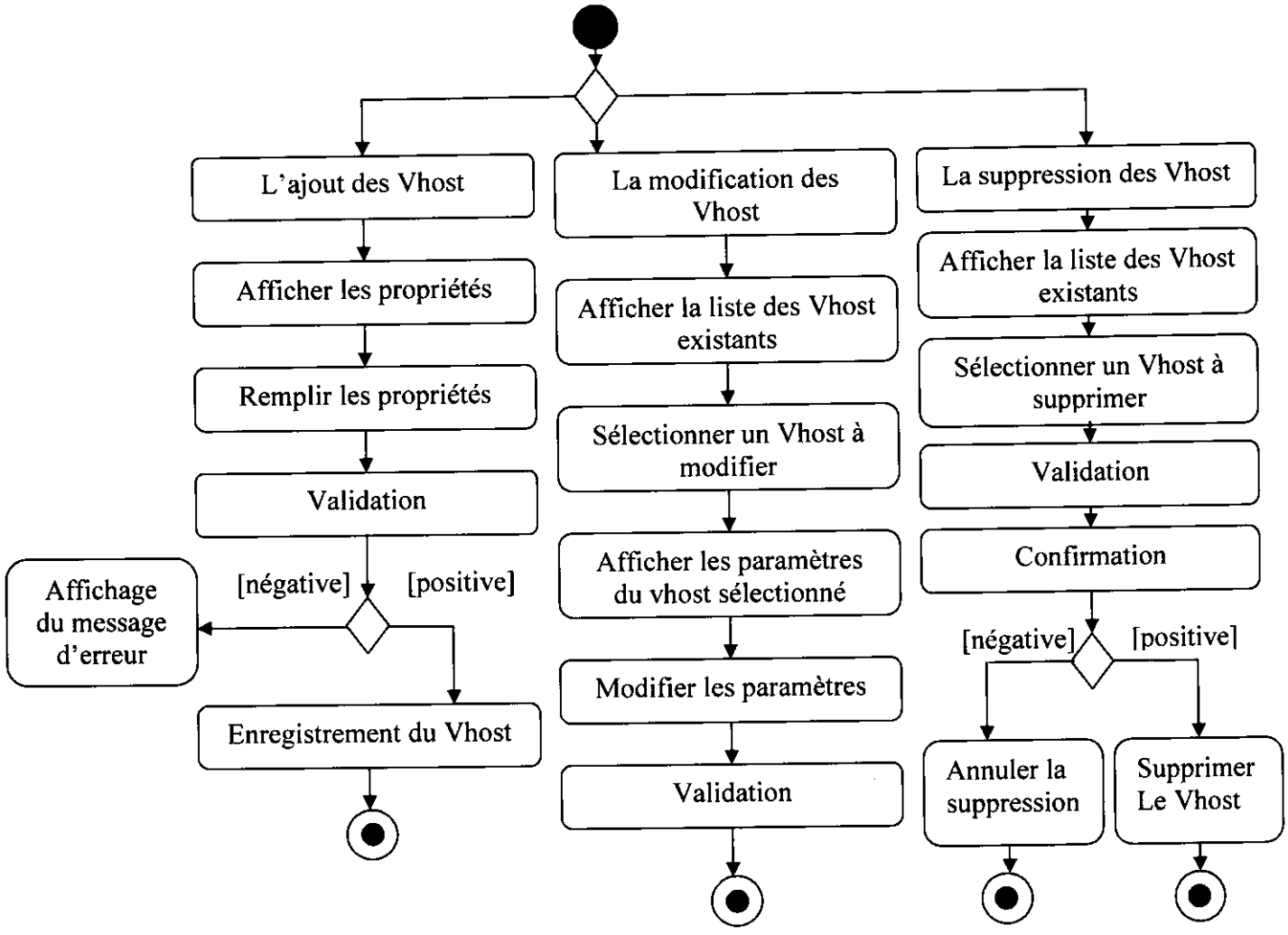
-Figure 3.9-

3_**Di**agramme d'activité de la protection des pages web :



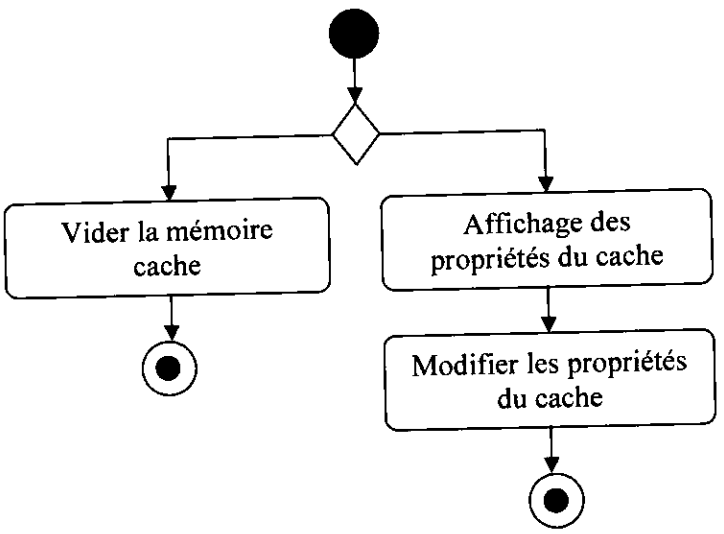
-Figure 3.10-

4_Diagramme d'activité de la gestion des Vhosts :



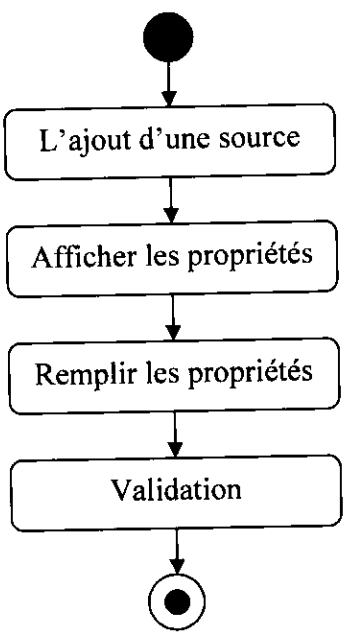
-Figure 3.11-

5_**D**iagramme d'activité pour la gestion de la mémoire cache de squid :



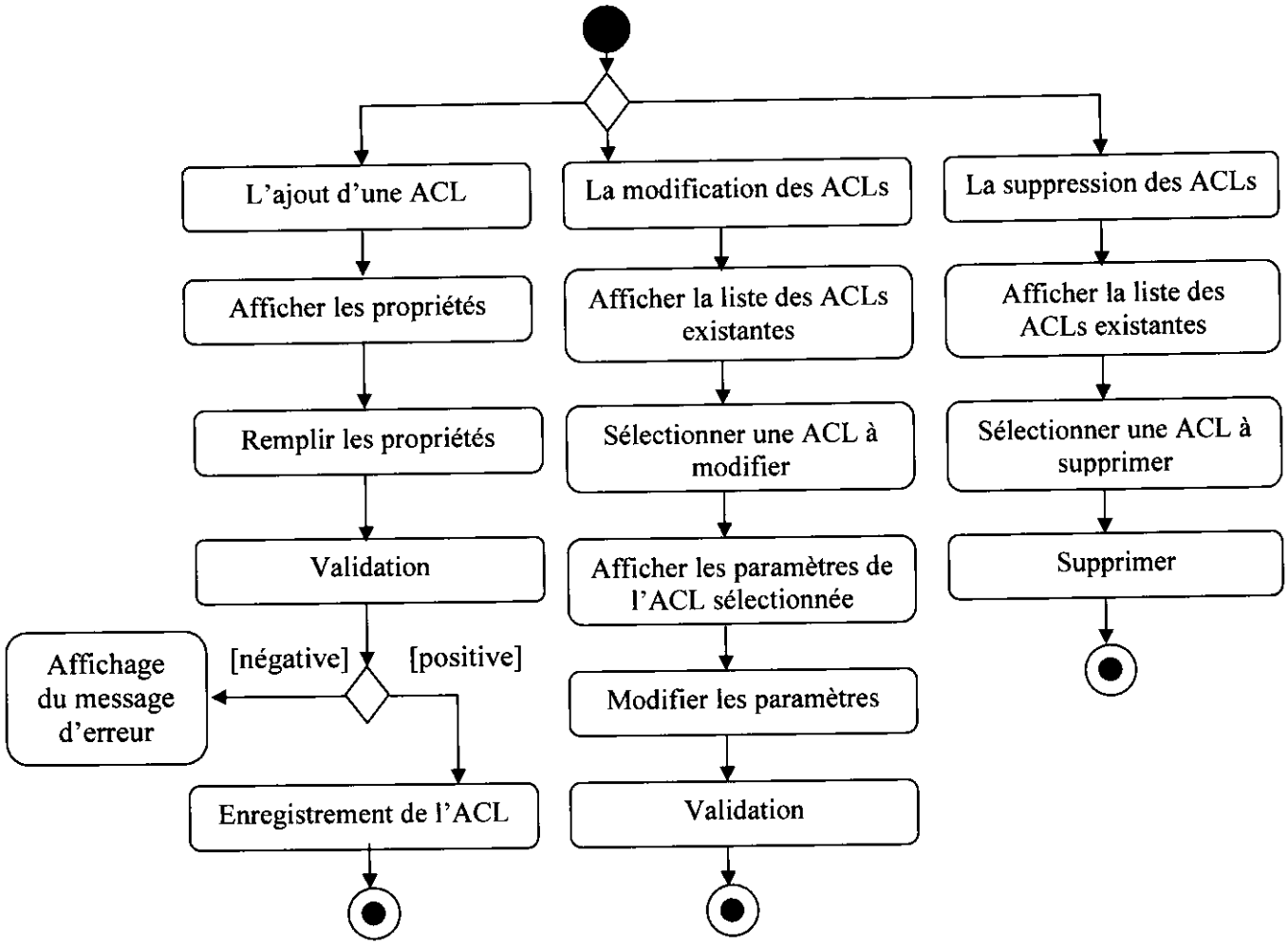
-Figure 3.12-

6_**D**iagramme d'activité pour l'ajout d'une source de squidGuard:



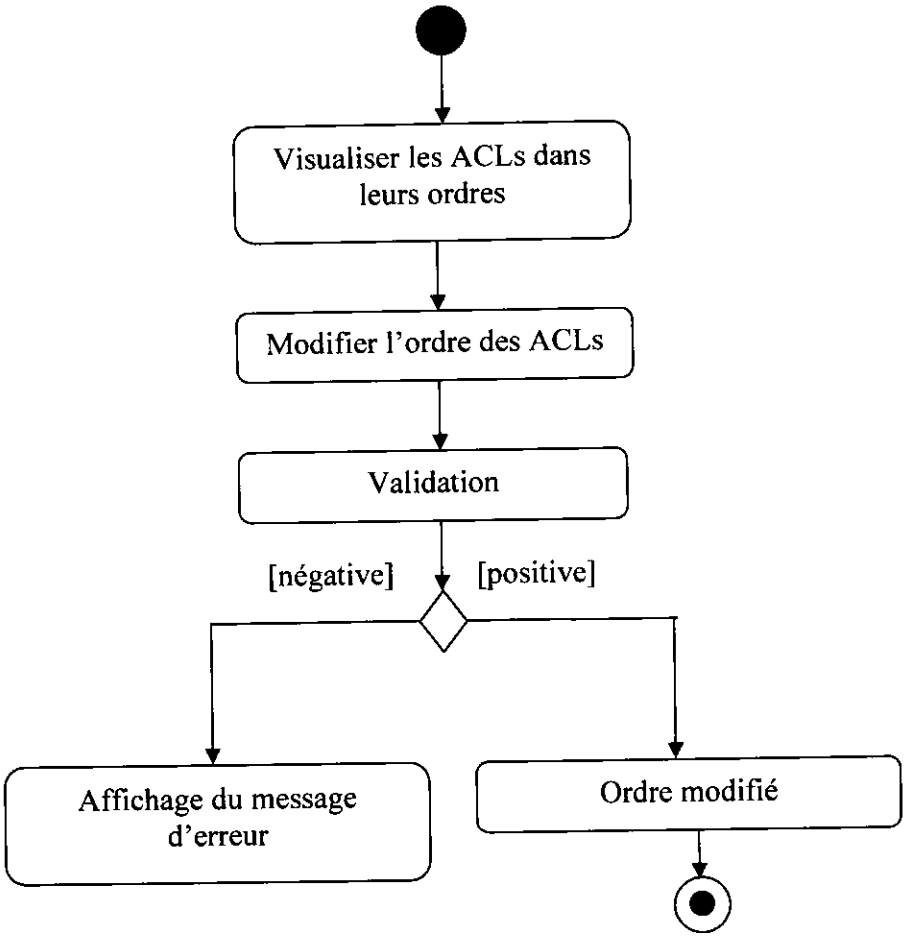
-Figure 3.13-

7_Diagramme d'activité pour la gestion des ACLs de squid :



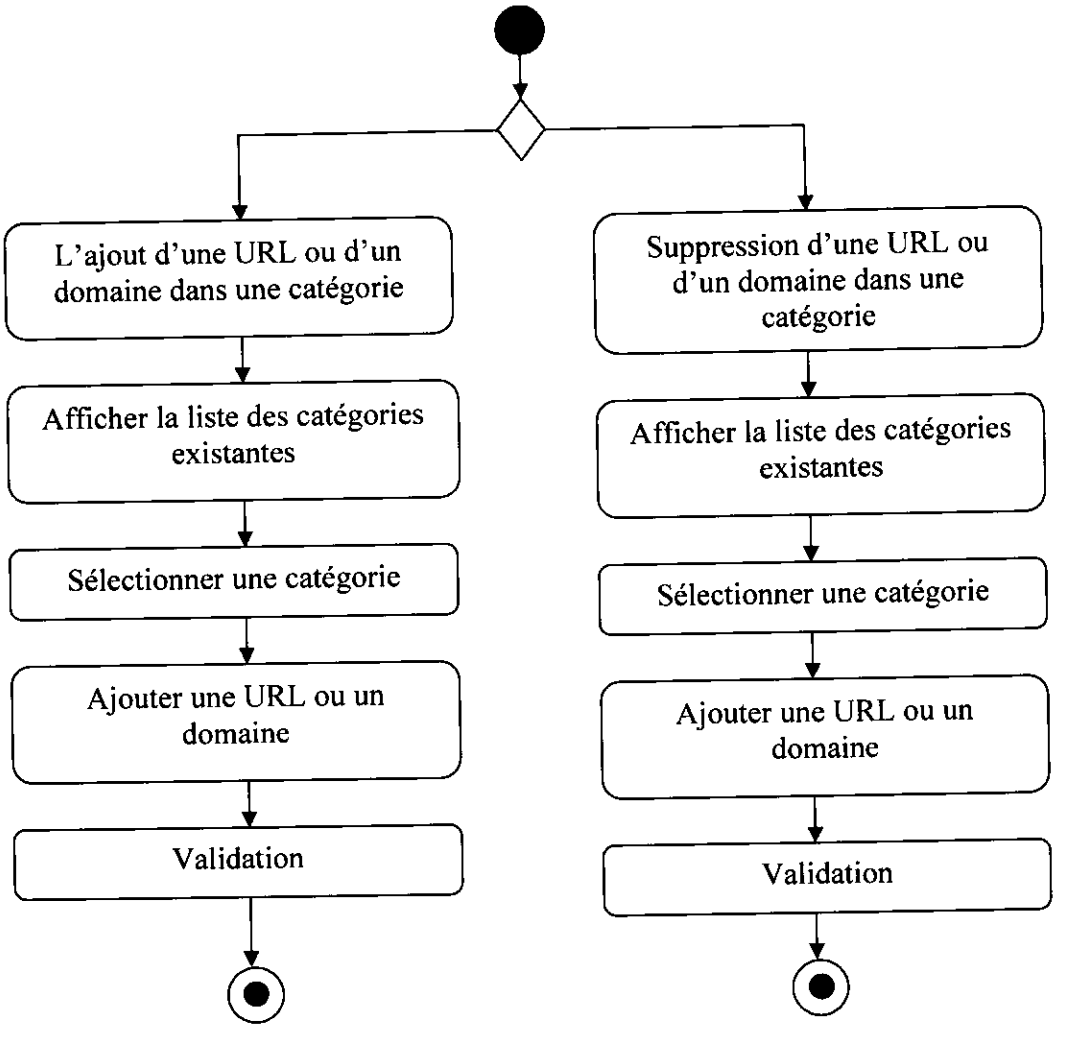
-Figure 3.14-

8_Diagramme d'activité pour la modification de l'ordre des ACL :



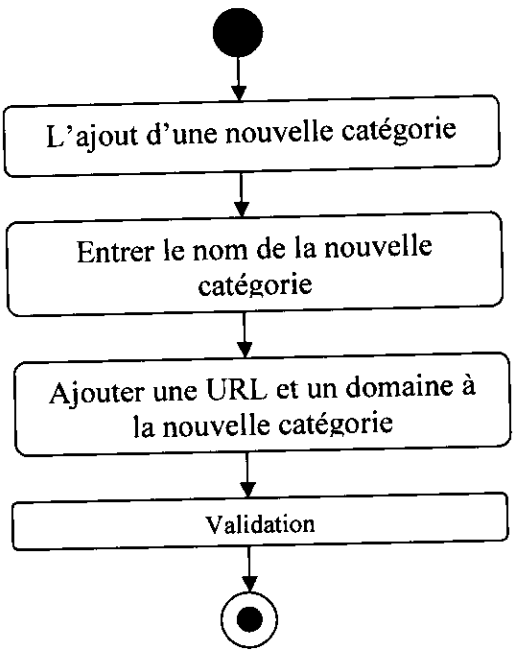
-Figure 3.15-

9_**Diagramme** d'activité pour l'ajout/suppression d'une URL ou un domaine dans une catégorie :



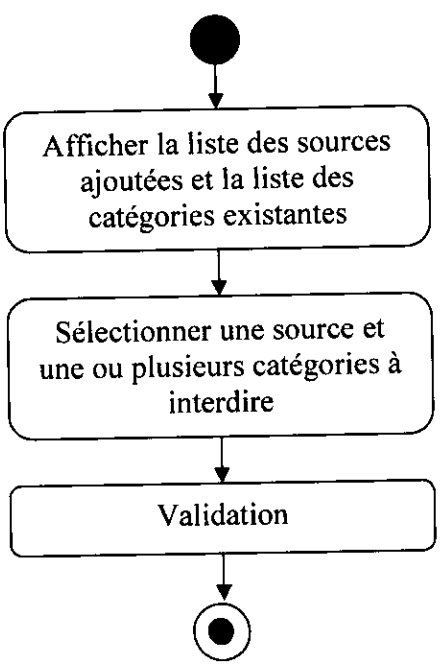
-Figure 3.16-

10_**D**iagramme d'activité pour l'ajout d'une nouvelle catégorie :



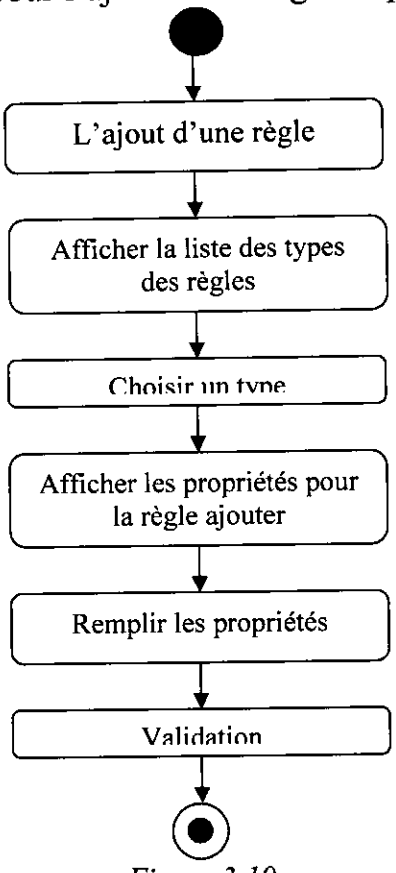
-Figure 3.17-

11_**D**iagramme d'activité pour l'édition du fichier de configuration de SquidGuard:



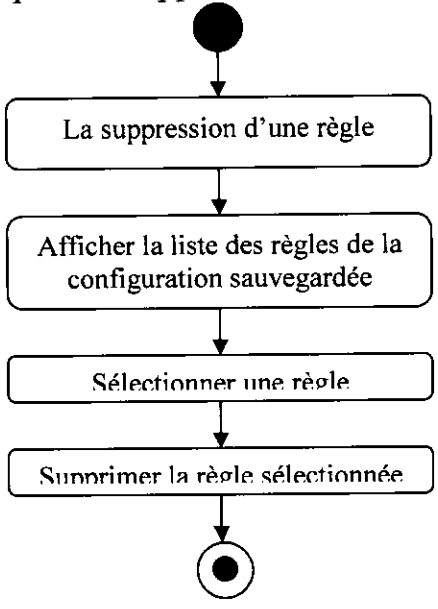
-Figure 3.18-

12_**D**iagramme d'activité pour l'ajout d'une règle d'Iptables :



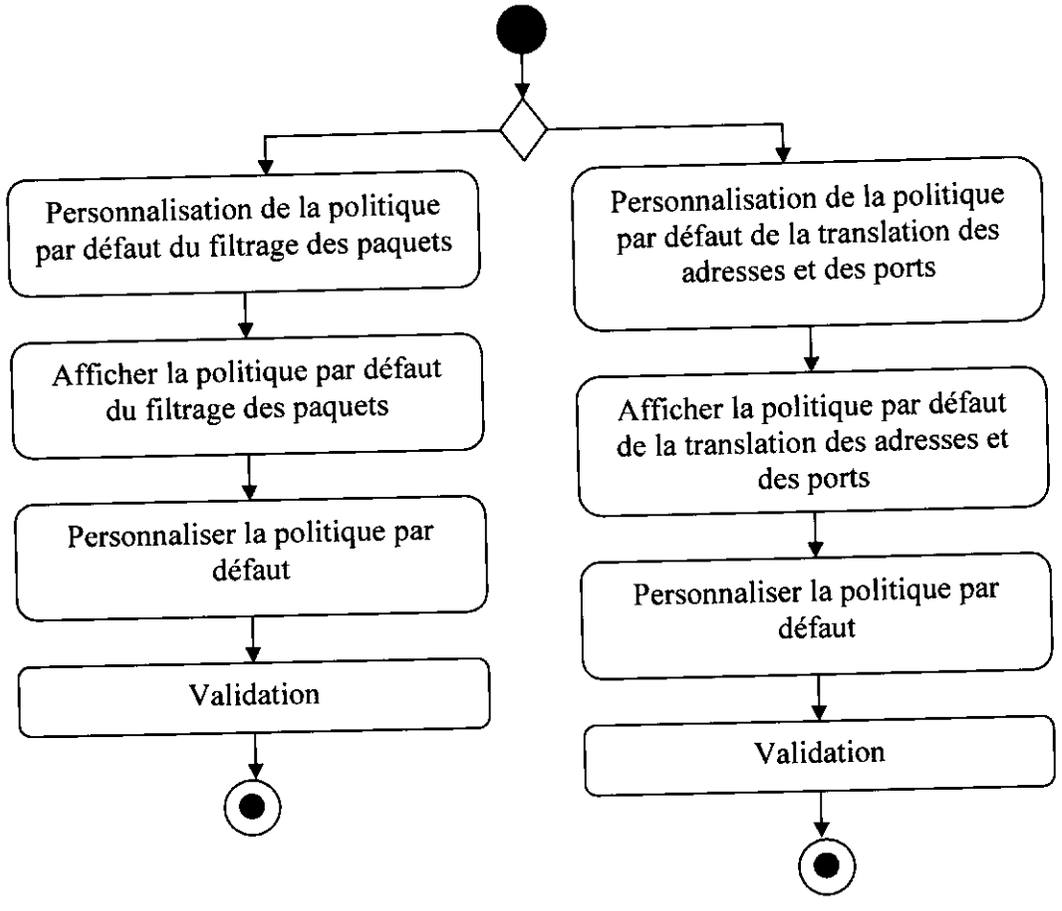
-Figure 3.19-

13_**D**iagramme d'activité pour la suppression d'une règle d'Iptables :



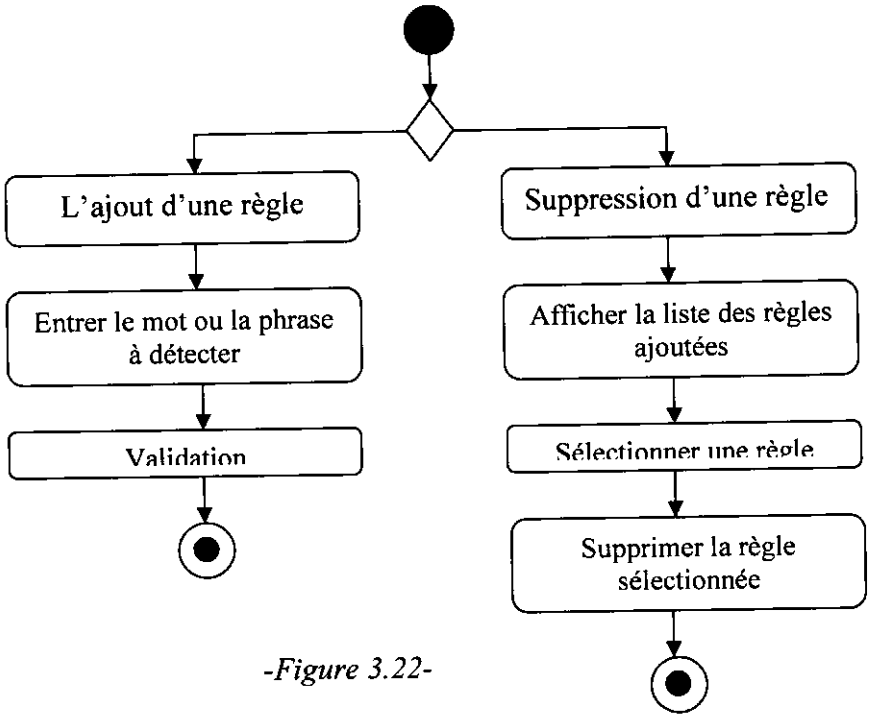
-Figure 3.20-

14_Diagramme d'activité pour la personnalisation de la politique par défaut d'Iptables :



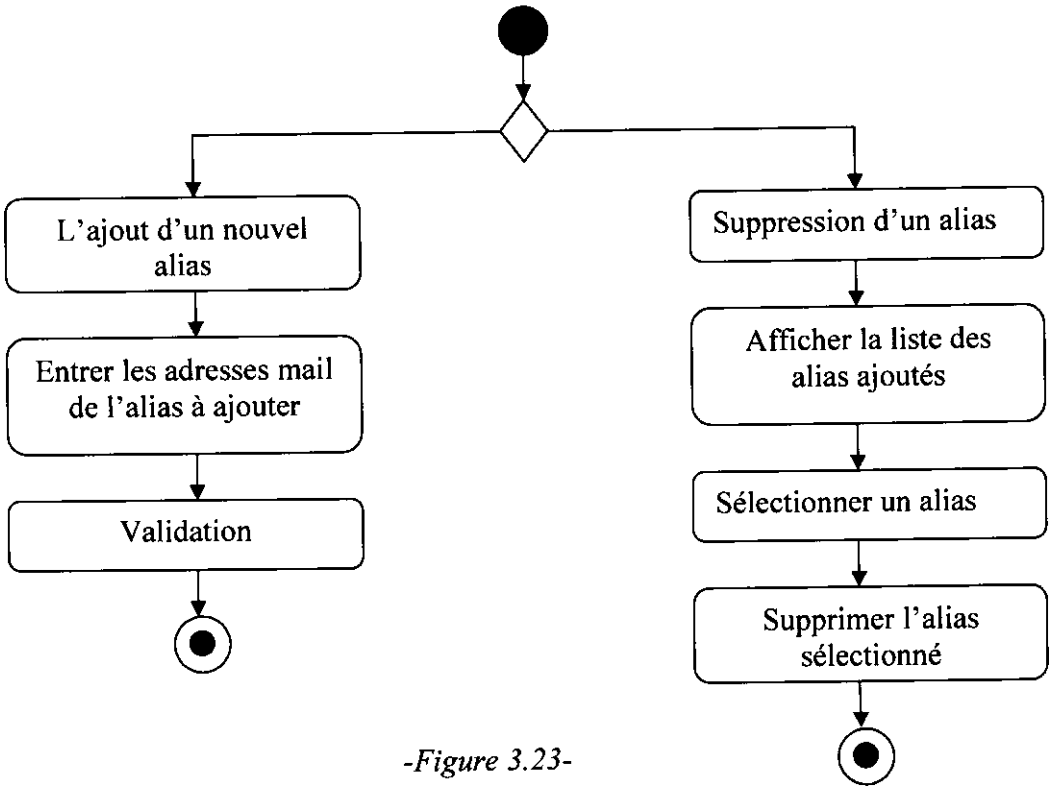
-Figure 3.21-

15_**D**iagramme d'activité pour la gestion des règles de filtrage des entêtes :



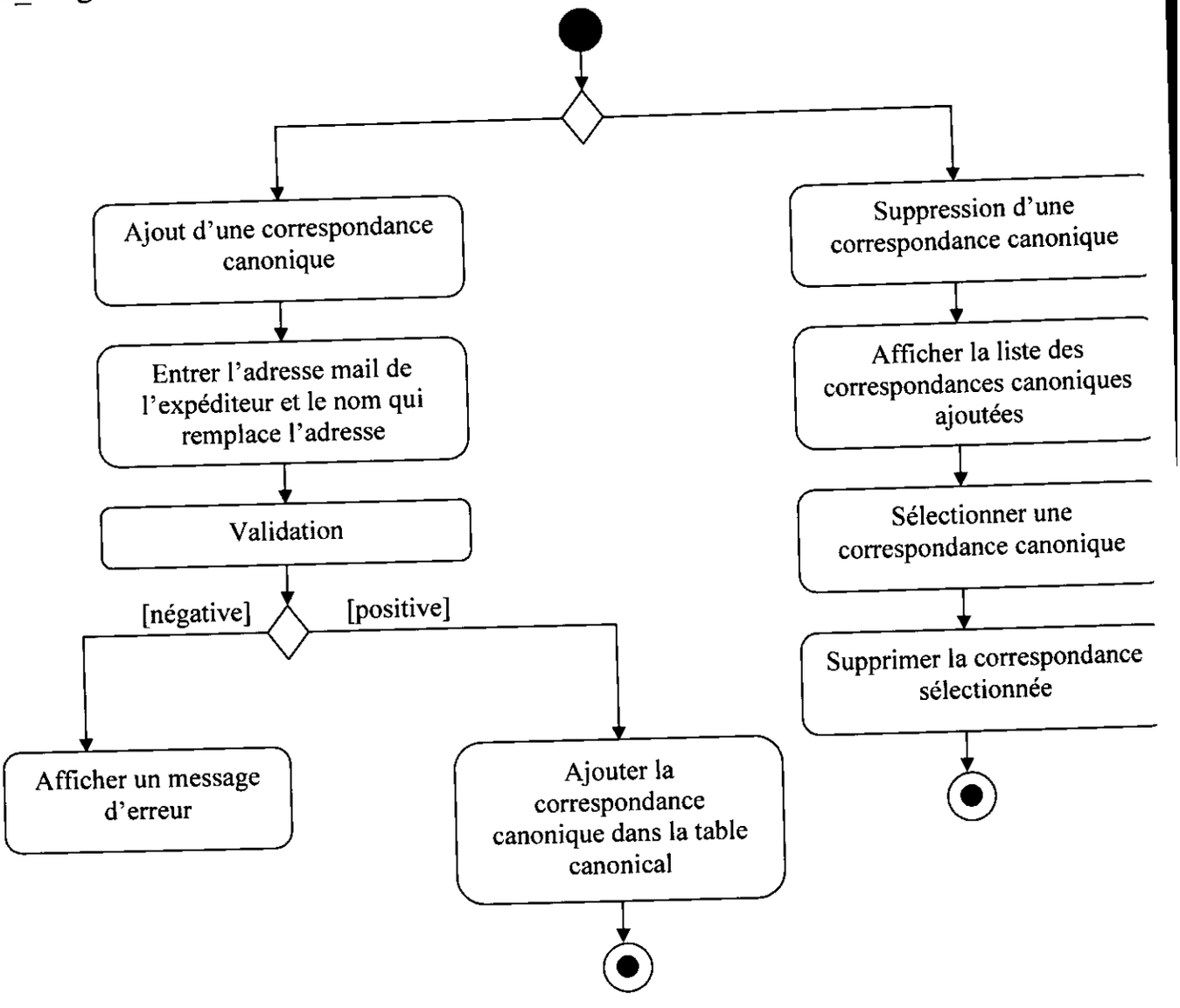
-Figure 3.22-

16_Diagramme d'activité pour la gestion de la table alias :



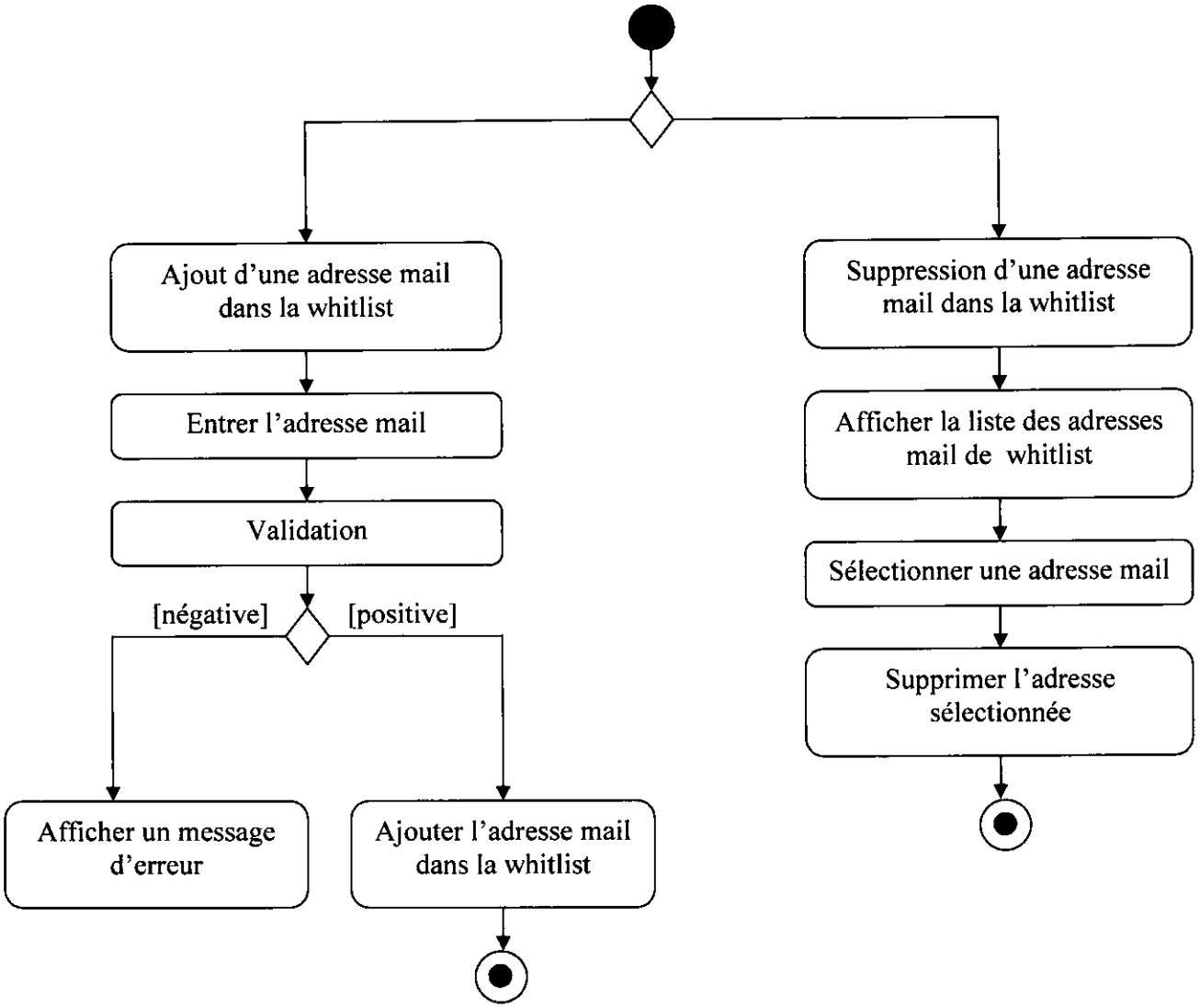
-Figure 3.23-

17_**D**iagramme d'activité pour la gestion des correspondances canoniques :



-Figure 3.24-

18_Diagramme d'activité pour la gestion de la whitelist (respectivement blacklist) de SpamAssassin :



-Figure 3.25-

III. Diagramme de séquence:

Un diagramme de séquence montre une interaction, c'est-à-dire un ensemble d'objets et leurs relations, ainsi que les messages qui peuvent circuler entre eux.

Un diagramme de séquence est un diagramme d'interaction qui met l'accent sur le classement des messages par ordre chronologique.

Nous prenons comme exemple le serveur web apache (c'est le même principe pour les autres services).

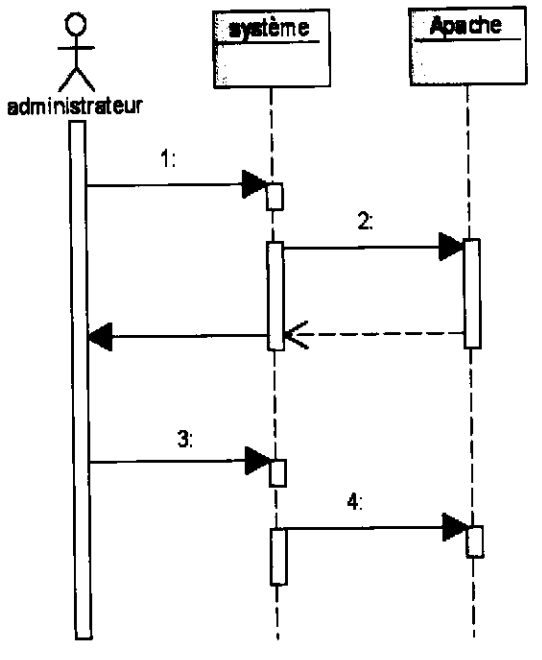
III.1.Diagramme de séquence pour la configuration de base d'Apache :

1 : l'utilisateur souhaite faire une configuration basique d'apache.

2 : le système consulte le fichier de configuration et affiche les propriétés de base d'Apache.

3 : l'administrateur modifie les propriétés de base et valide.

4 : le système prend en charge la nouvelle configuration en modifiant le serveur apache.



-Figure 3.26-

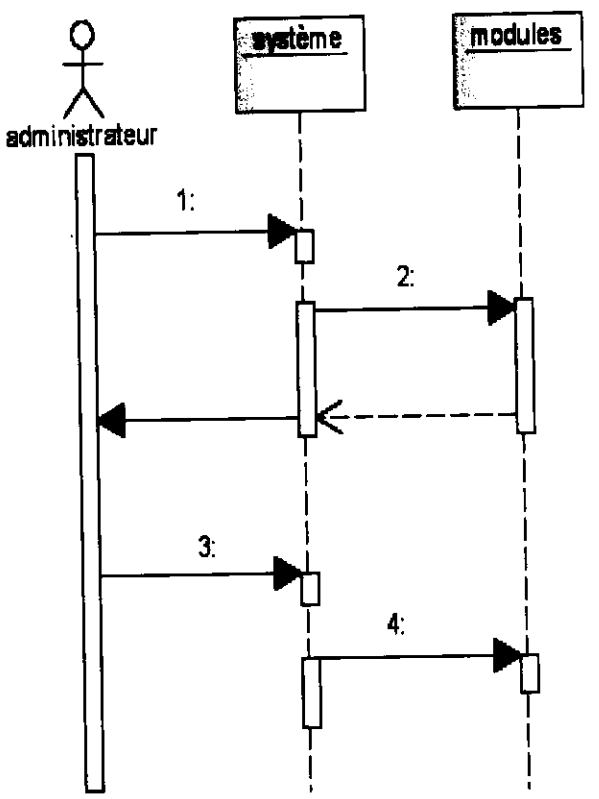
III.2.Diagramme de séquence pour chargement/déchargement des modules :

1 : l'utilisateur souhaite configurer les modules d'apache.

2 : le système affiche tous les modules d'apache selon leurs états (chargé ou déchargé).

3 :l'administrateur sélectionne des modules à charger et désélectionne les modules à décharger, et valide.

4 : le système charge les modules sélectionnés et décharge les modules désélectionnés.



-Figure 3.27-

III.3. Diagramme de séquence pour la modification d'un vhost :

1 : l'utilisateur souhaite faire une modification d'un vhost.

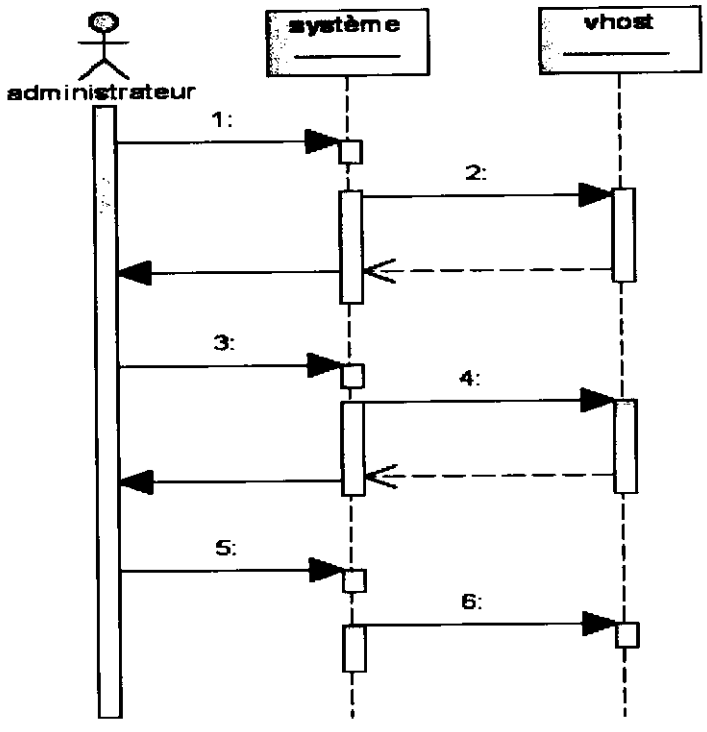
2 : le système affiche la liste des vhost existants.

3 : l'administrateur sélectionne le vhost à modifier.

4 : le système affiche les propriétés du vhost sélectionné.

5 : l'administrateur modifie les propriétés et valide.

6 : le système prend en charge la nouvelle.



-Figure 3.28-

Chapitre 4 :

Réalisation & tests

Chapitre 4

Réalisation & tests

Au cours des chapitres précédents de ce travail, nous avons détaillé notre système : les services réseaux à configurer, leurs fichiers de configuration et les commandes d'administration. Nous arrivons à présent à la dernière partie du développement où nous allons présenter la phase réalisation.

I) L'environnement de développement matériel :

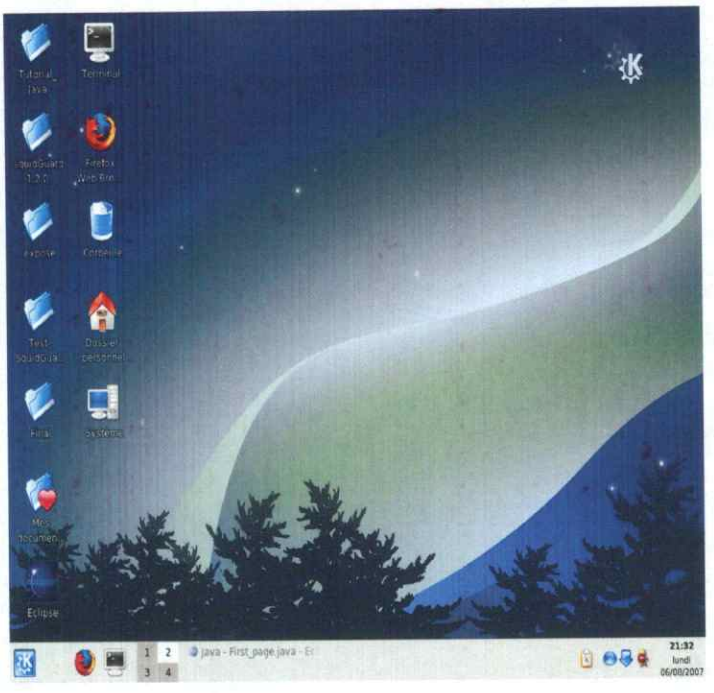
La réalisation de notre outil a nécessité un pc doté d'une puissance suffisante pour l'installation de linux et les services réseaux vus dans le **chapitre II**.
Pour l'utilisation, toute machine dans la catégorie Pentium 4 avec une **RAM** 512 **MO**, deux cartes réseaux une reliée au réseau local et l'autre à Internet fera parfaitement l'affaire.
A noter que cela dépend bien évidemment de la fréquence d'utilisation, du nombre de clients, du débit etc. Et plus sont les ressources d'**UC**, meilleur est le résultat.

II) L'environnement de développement logiciel :

II-1) Le système d'exploitation :

Nous avons choisi Fedora Core 6 qui est une distribution financée et éditée par la société RedHat avec l'aide de la communauté des utilisateurs de logiciels libres.

Elle est disponible en libre téléchargement sur un grand nombre de ftp **[WWW_7]**



-Figure 4.1-

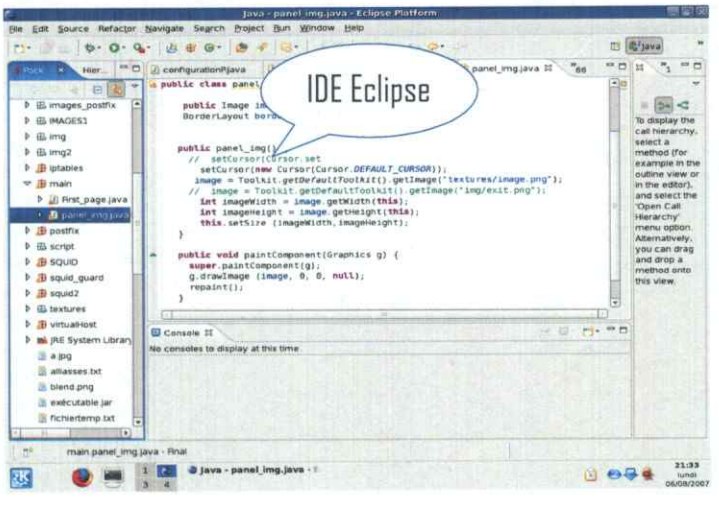
II-2) Le langage de programmation :

Notre choix s'est porté sur le langage JAVA. Rappelons que Fedora dispose d'origine d'un JAVA entièrement libre, élément de la collection des compilateurs GNU **[WWW_8]**.

Ce choix se justifie par les raisons suivantes :

- JAVA est un langage orienté objet simple et sûr (un programme Java planté ne menace pas le système d'exploitation).
- JAVA assure la gestion de la mémoire : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au « garbage collector » qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- JAVA possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standard, le système de gestion de fichiers, gestion de processus et beaucoup d'autres fonctionnalités **[L _2]**.

Nous avons conçu notre application java en utilisant l'environnement de développement Eclipse **[WWW_11]** qui est un environnement de développement intégré. Eclipse offre de nombreuses fonctionnalités. Il comporte notamment une série d'outils permettant le développement, l'exécution, le débogage des applications java.



-Figure 4.2-

III) Problème et solutions proposées :

Au cours de la phase de programmation, nous avons rencontré les problèmes suivants :

- Notre outil fonctionne avec les privilèges root, cela signifie qu'il est tout à fait possible de détruire le système en cas de manipulation incorrecte, afin d'éviter cet éventuel problème nous avons gardé des copies des fichiers de configuration en bon état des divers services, et c'est à l'utilisateur de les restaurer en cas de nécessité.

- il comporte également un module de création de serveur virtuel, il est donc possible de créer à partir du programme un répertoire qui va servir comme répertoire racine du serveur virtuel dans ce cas et pour éviter d'avoir un répertoire racine vide le programme génèrera une page d'accueil par défaut pour chaque répertoire créé.

- L'ordre des ACLs et très important, car la première ACL qui correspond aux critères sera applicable et les autres seront ignorées. Pour y remédier nous avons implémenté un module « Ordre des acls » et l'utilisateur pourra à tout moment changer l'ordre de ses ACLs selon ses besoins.

- La blacklist officielle (par défaut avec SquidGuard) n'étant pas complète nous avons réalisé un module de Personnalisation de blacklist et l'utilisateur pourra soit en ajouter (ou supprimer) des domaines ou des urls, ou bien créer une nouvelle catégorie (par exemple l'ajout de la catégorie chat qui va lister les salons de chat).

- Il se peut qu'un administrateur expert ajoute manuellement (sans passer par nôtre outil) une catégorie dans la blacklist, cette catégorie sera prise en compte même si elle n'a pas été créée via le programme.

- Les règles d'iptables générées à partir de nôtre application seront automatiquement sauvegardées, mais si un administrateur expert ajoute une ou plusieurs règles à partir de l'interpréteur de commandes il pourrait les sauvegardées (nous avons ajouté une option dans le programme).

- Par défaut, les règles d'iptables ne sont sauvegardées que pour la session en cours, mais suite à une demande d'utilisateur nôtre outil propose de lancer le script d'iptables à chaque démarrage du système.

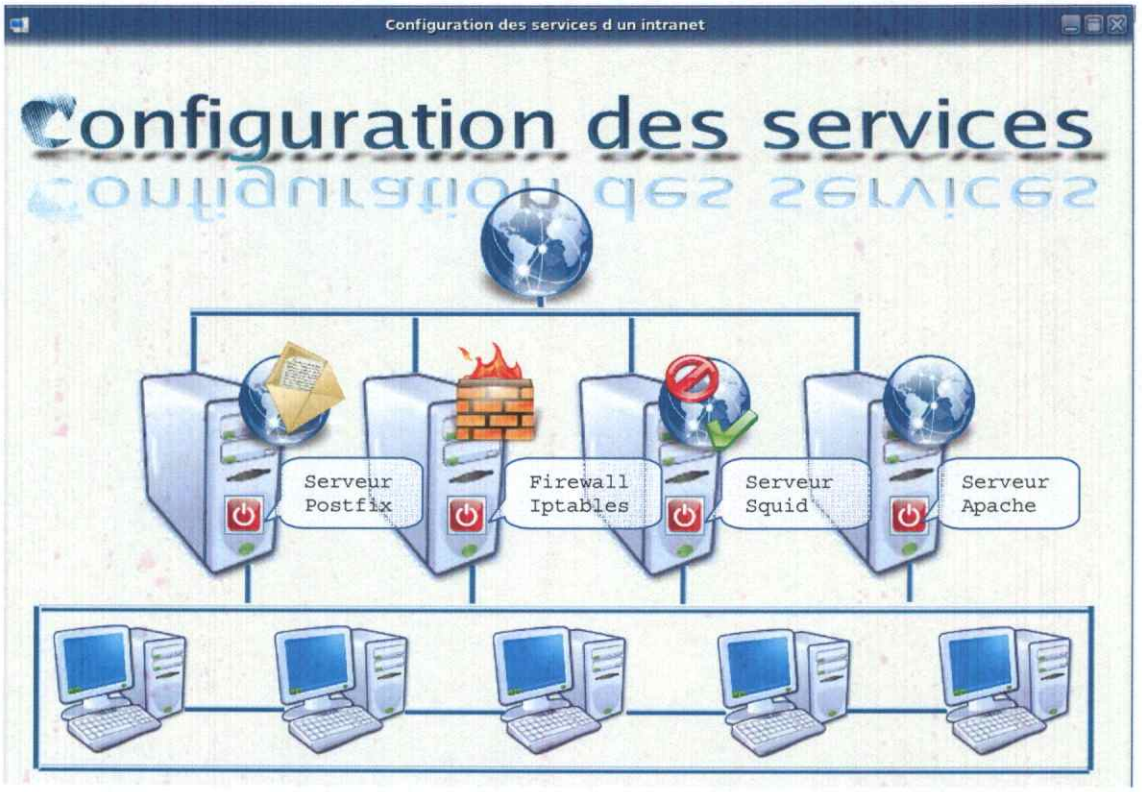
IV) Présentation de l'interface utilisateur :

Afin de simplifier le travail d'un administrateur, nous avons conçu une interface simple et ergonomique.

Nous présentons dans cette partie son mode d'utilisation accompagné des bulles d'information ainsi que les principaux tests.

- L'interface principale:

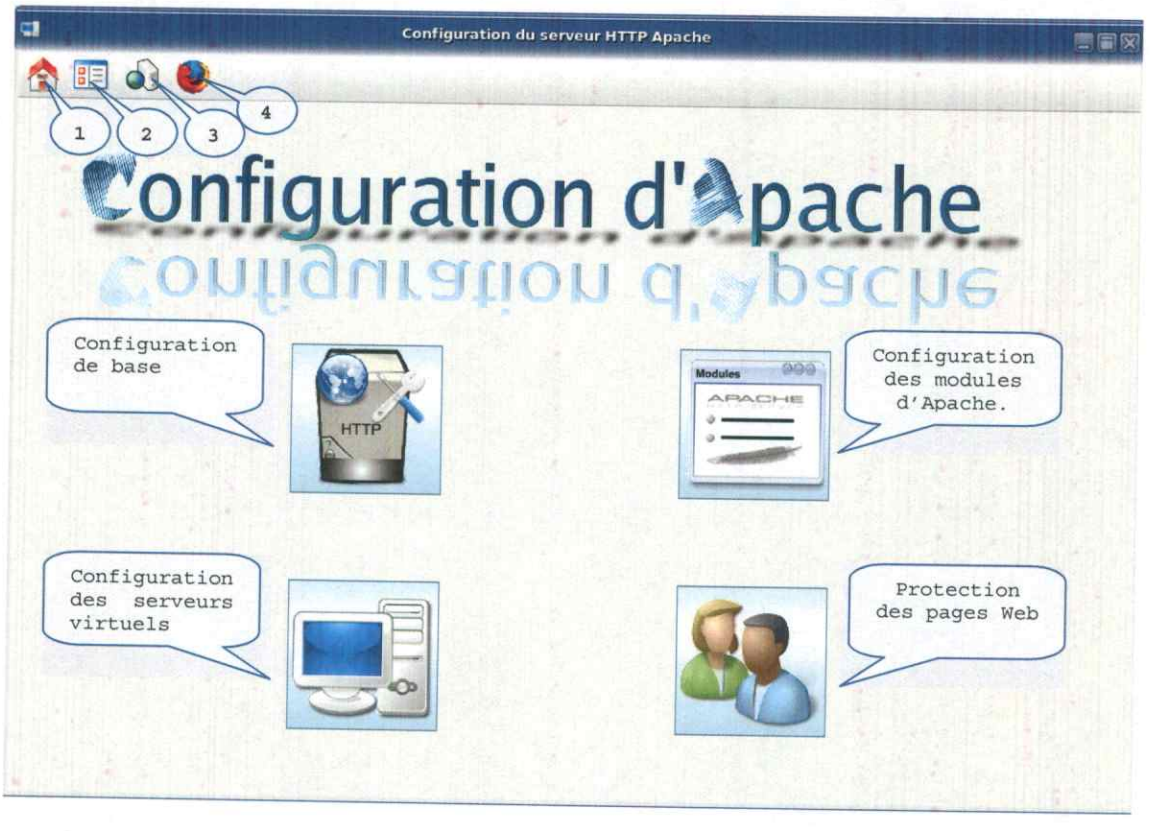
C'est la page d'accueil de notre Outil, c'est à partir de cette page que l'utilisateur pourra lancer la configuration des différents services :



-Figure 4.3-

- L'interface de configuration d'Apache :

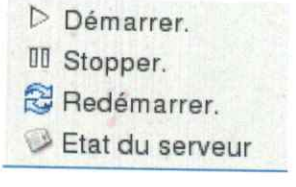
La figure ci-dessous montre l'interface de configuration du serveur Apache. Cette interface se compose principalement de 4 modules de configurations ainsi qu'un Menu.



-Figure 4.4-

Le Menu : Se compose de 4 items

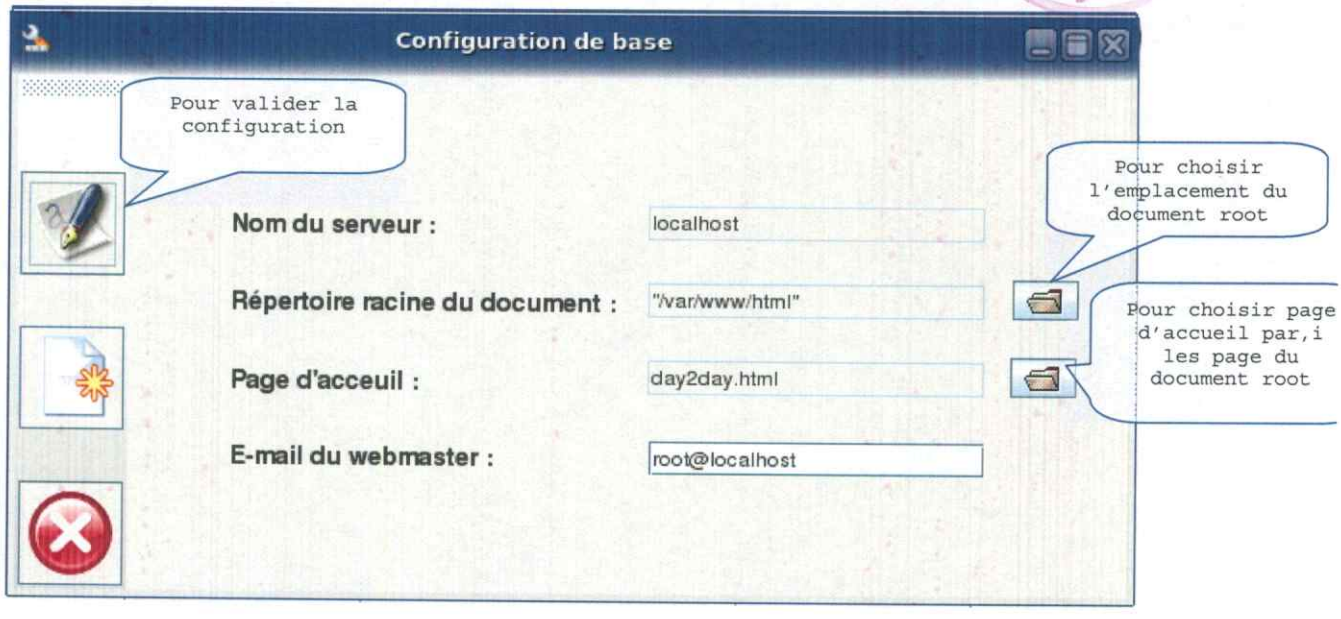
- L'item **N°=1** : pour revenir à la page d'accueil.
- L'item **N°=2** : pour voir le fichier de configuration.
- L'item **N°=3** : pour démarrer, arrêter, redémarrer et voir l'état du serveur.



- L'item **N°=4** : pour ouvrir le navigateur FireFox.

- Interface de configuration de base d'Apache :

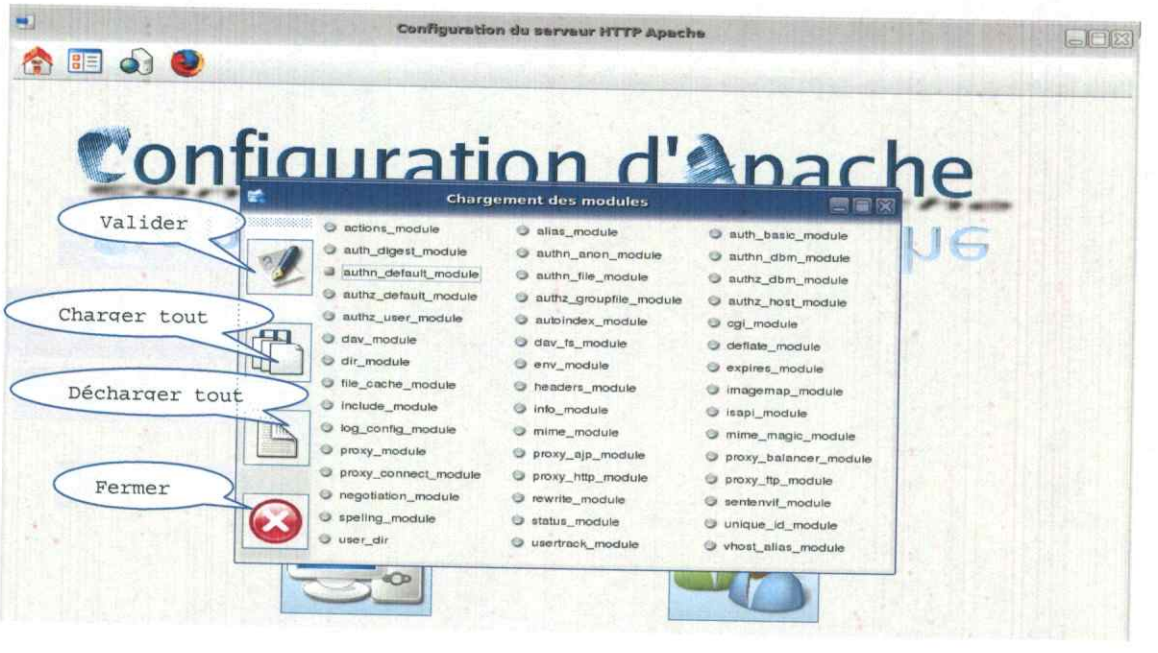
Pour changer les paramètres de base du serveur Apache :



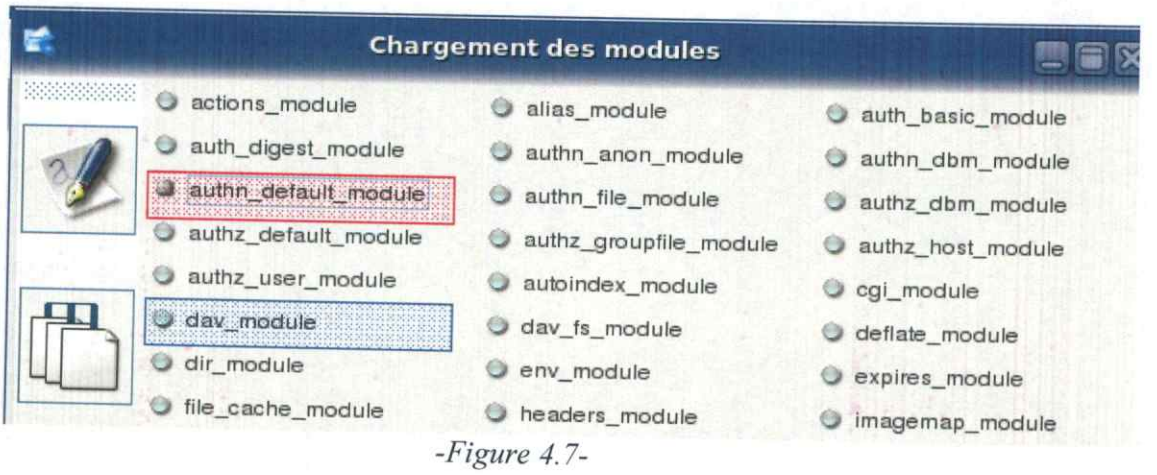
-Figure 4.5-

- L'interface de configuration des modules :

L'utilisateur peut charger ou décharger un module selon ses besoins, il suffit juste de cocher ou décocher le module approprié. Les modules sont affichés selon leur état dans le fichier de configuration d'Apache.



-Figure 4.6-



-Figure 4.7-

Module déchargé.

Module chargé

- Interface de protection des pages web :

Notre outil facilite la protection des pages. Il suffit d'entrer l'emplacement de la page ainsi que le nom d'utilisateur suivi du mot de passe avec sa confirmation.

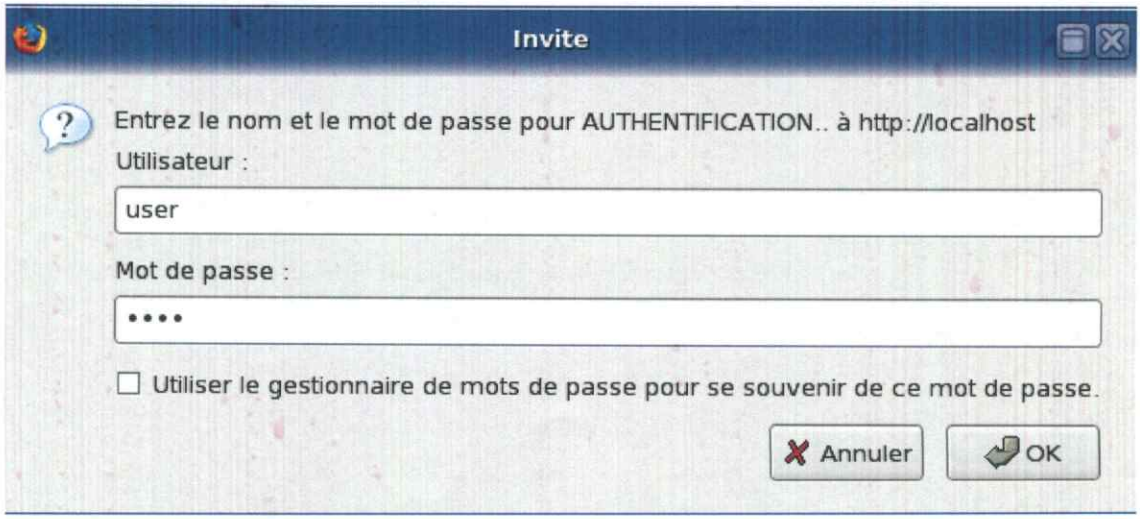
Note : il est possible d'autoriser plusieurs utilisateurs pour la même page (refaire la même opération pour chaque utilisateur).

-Figure 4.8-

Test :

Dans le navigateur si l'utilisateur tape ***localhost***, il aura une boite d'Authentification et seuls les utilisateurs authentifiés auront l'accès à la page.

.

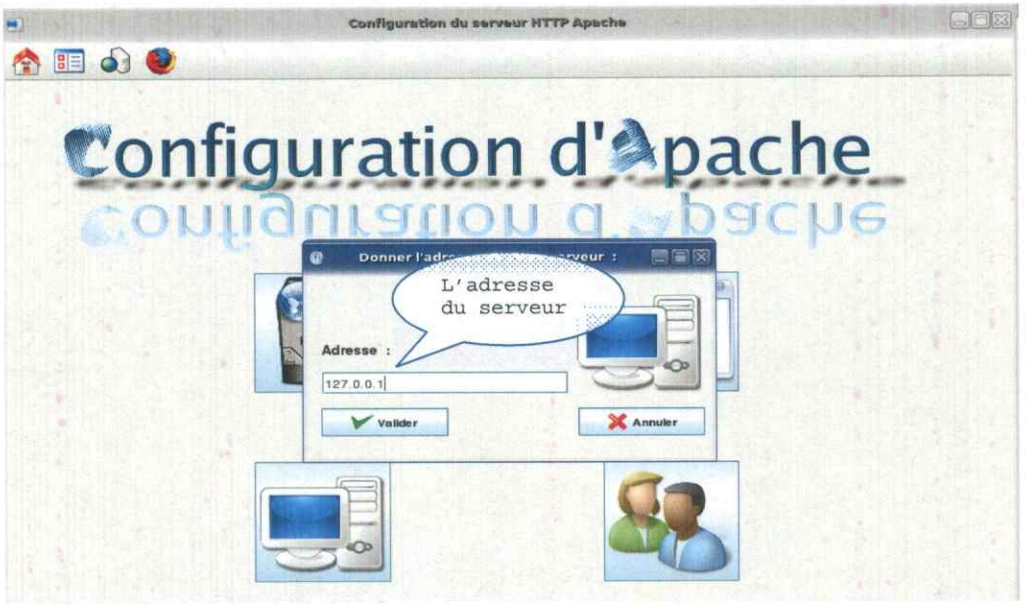


-Figure 4.9-

- Interface de configuration des serveurs virtuels :

Notre outil facilite la gestion des serveurs virtuels (création, modification et suppression).

Au début, il faut que l'utilisateur entre l'adresse du serveur virtuel.

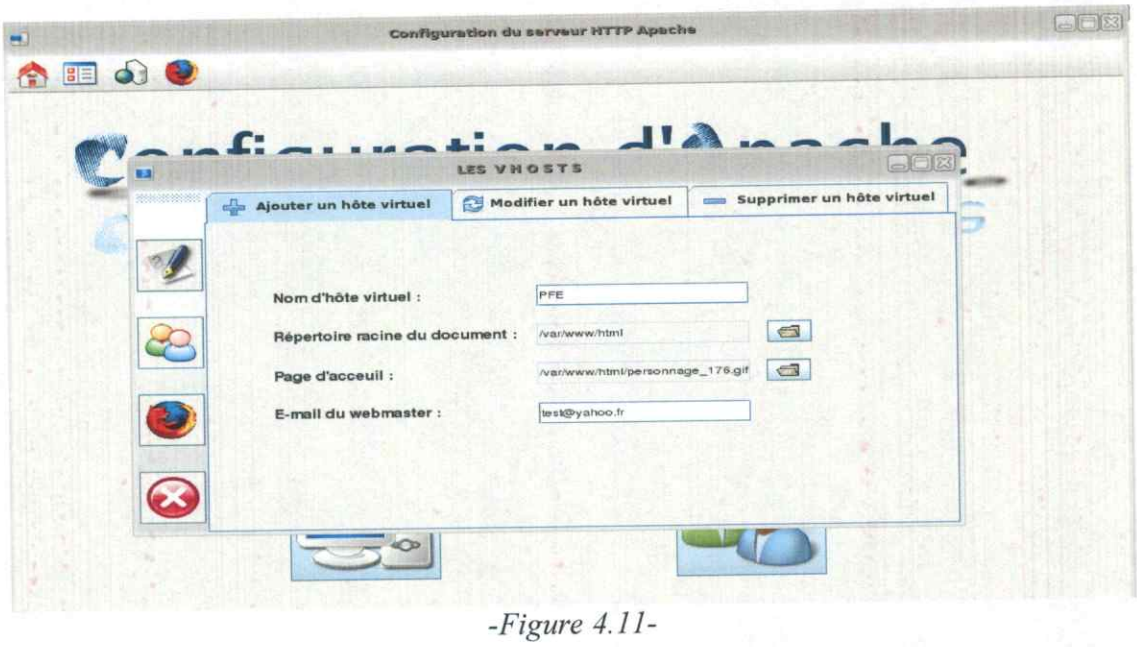


-Figure 4.10-

Ensuite il doit sélectionner l'opération désirée (Ajout, suppression ou modification). La section suivante explique les détails de chaque opération.

a) L'interface de création d'un serveur virtuel :

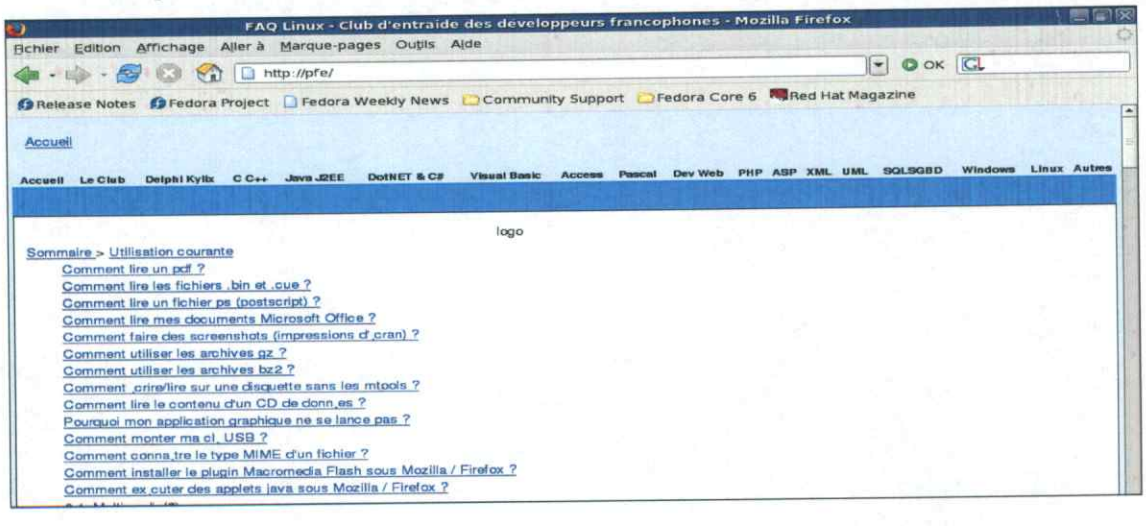
Pour cela il faut entrer le nom du serveur virtuel, l'emplacement de son répertoire racine, une page du répertoire racine qui va lui servir comme page d'accueil et l'email du Web master et à la fin valider la création.



-Figure 4.11-

Test :

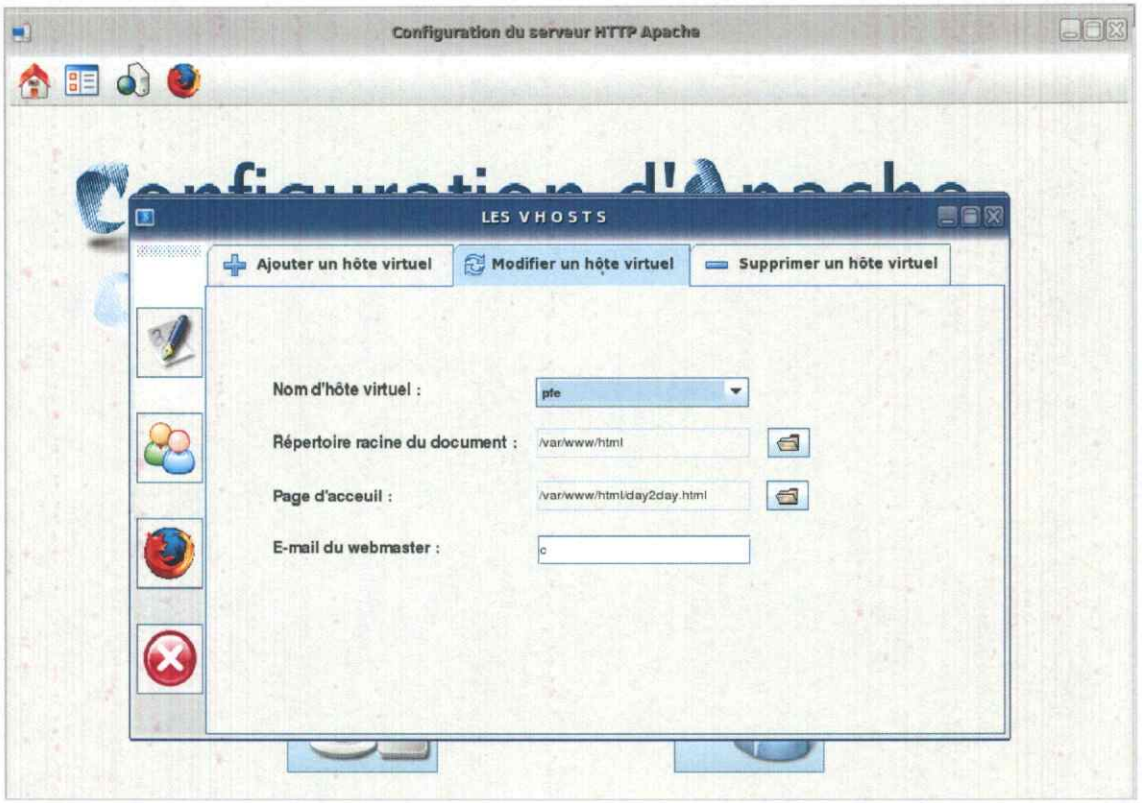
Maintenant pour tester, l'utilisateur doit entrer le nom du vhost.



-Figure 4.12-

b) L'interface de modification d'un serveur virtuel :

Il est possible de modifier les paramètres d'un serveur virtuel qui existe, prenons l'exemple précédent.



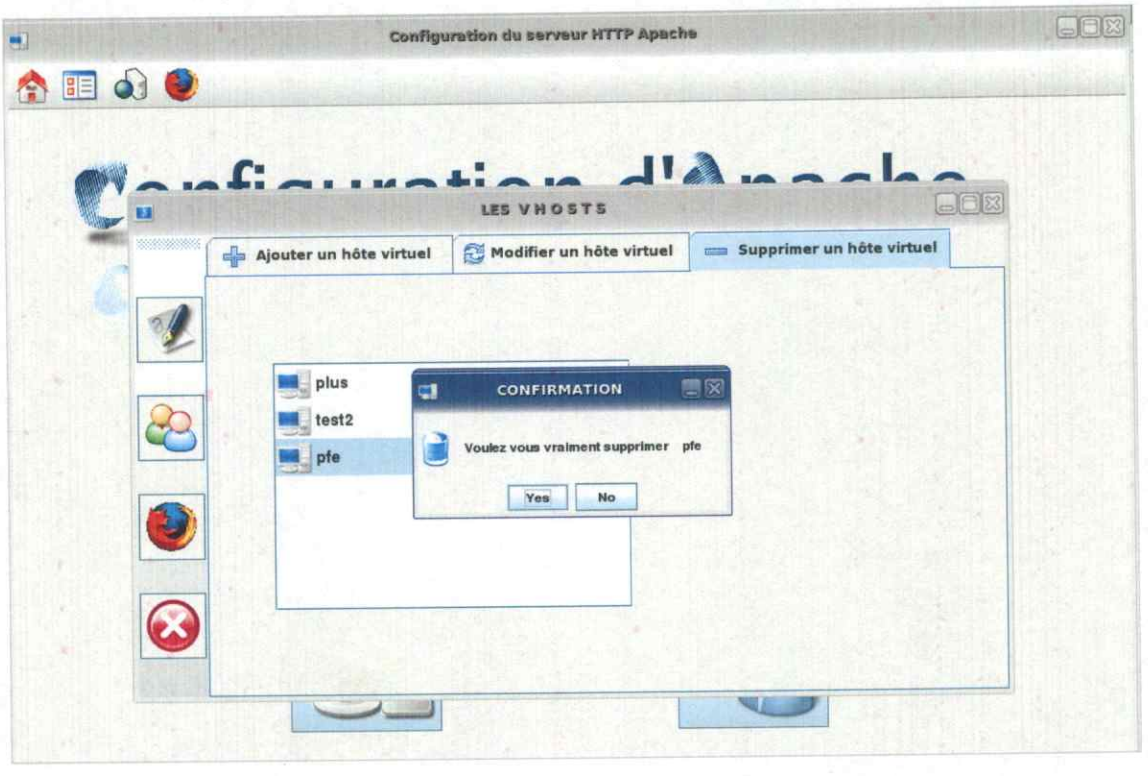
-Figure 4.13-

Test :

-Figure 4.14-

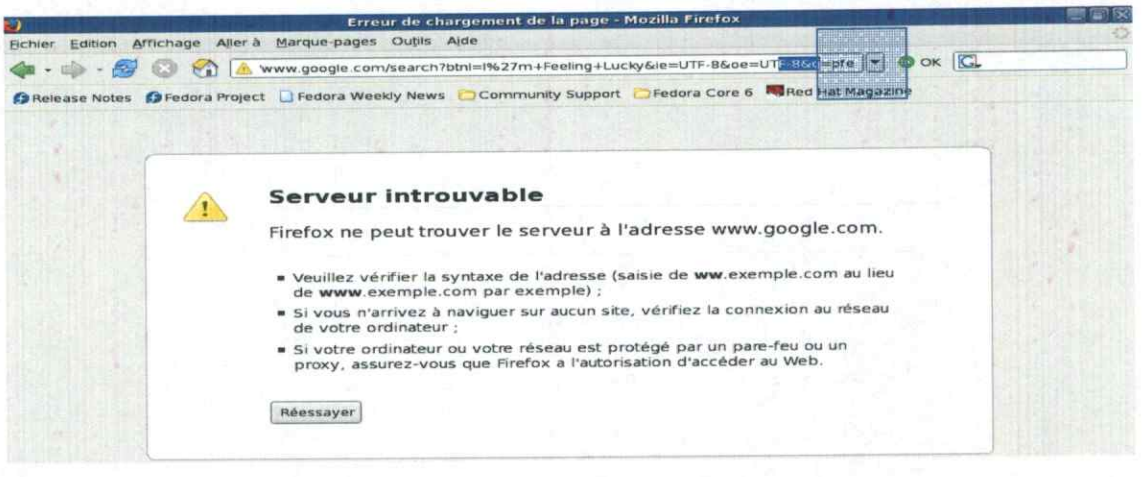
c) L'interface de suppression d'un serveur virtuel :

L'utilisateur n'a qu'à sélectionner le vhost qu'il veut supprimer.



-Figure 4.15-

Test :



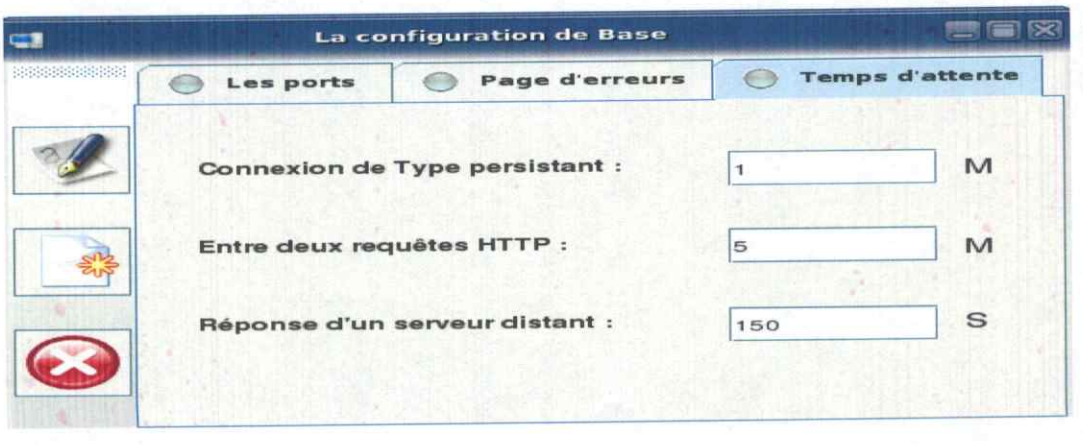
-Figure 4.16-

- L'interface de configuration de Squid :

-Figure 4.17-

- La configuration de base de Squid :

a) Le temps d'attente

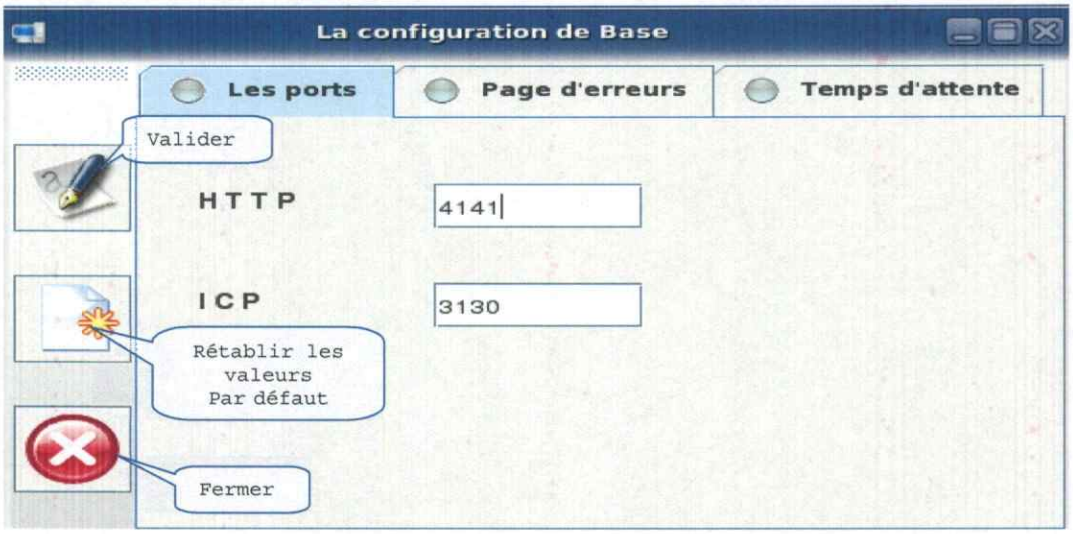


-Figure 4.18-

b) Les ports

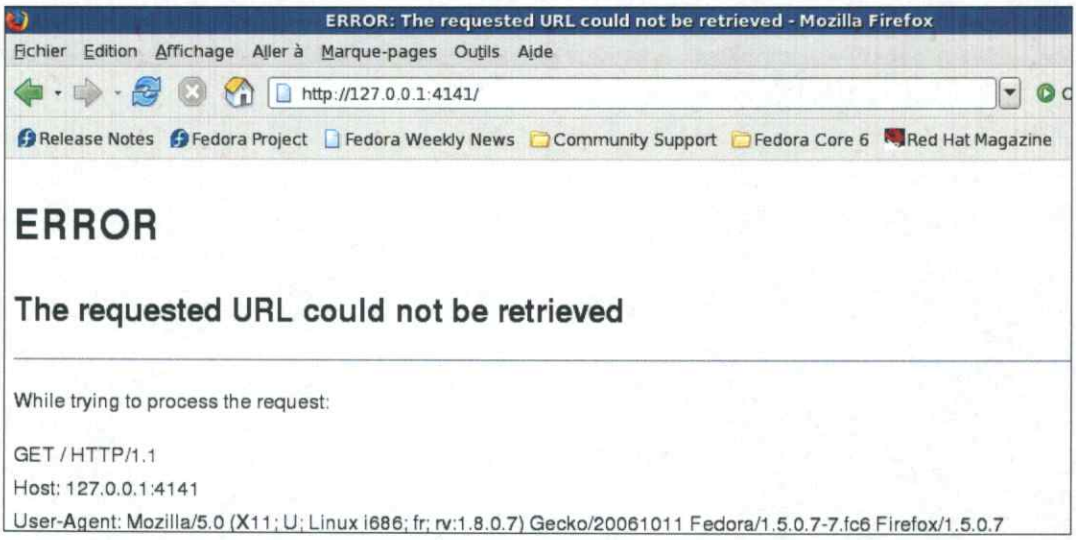
Pour changer les ports c'est simple, il suffit d'indiquer la valeur souhaitée et valider.

Dans cet exemple nous allons changer le port dont squid écoute (http Port):



-Figure 4.19-

Test :



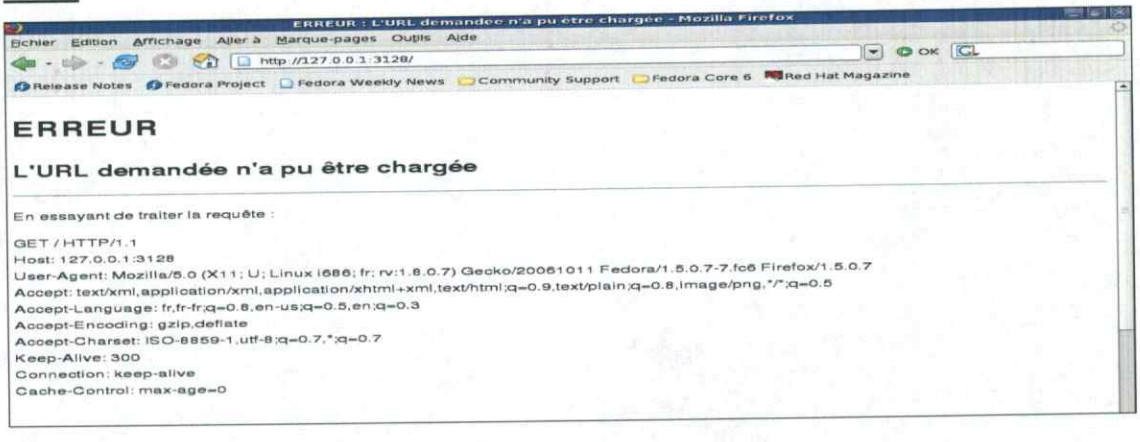
-Figure 4.20-

c) Page d'erreur :

Les pages d'erreurs de Squid sont par défaut en anglais. Notre outil permet de changer la langue ainsi que le nom de la machine visible et l'administrateur du cache.

-Figure 4.21-

Test :

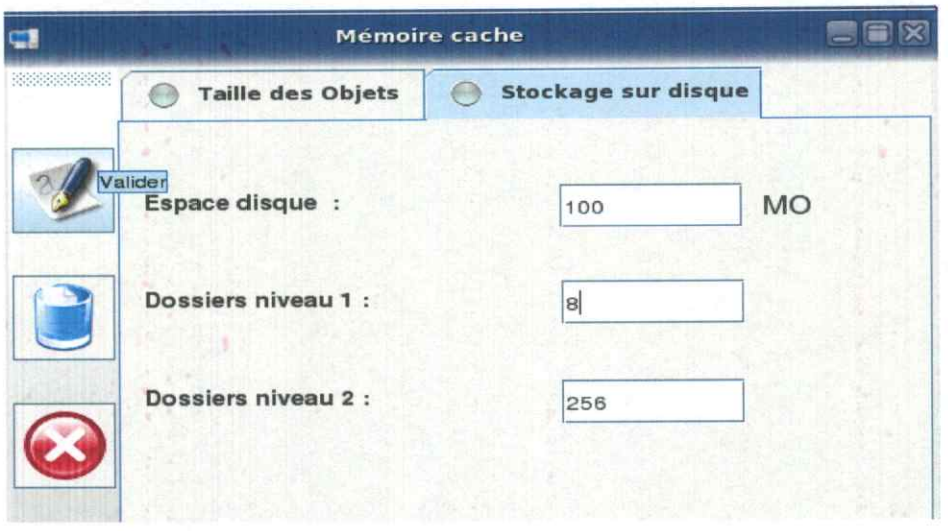


-Figure 4.22-

- Hiérarchie du cache :

Il est possible de changer la taille sur disque de la mémoire cache ainsi que le nombre de dossiers de niveau un et de niveau 2.

Dans cet exemple « le nombre de répertoire de niveau 1 » = 8 et 256 pour le niveau 2.

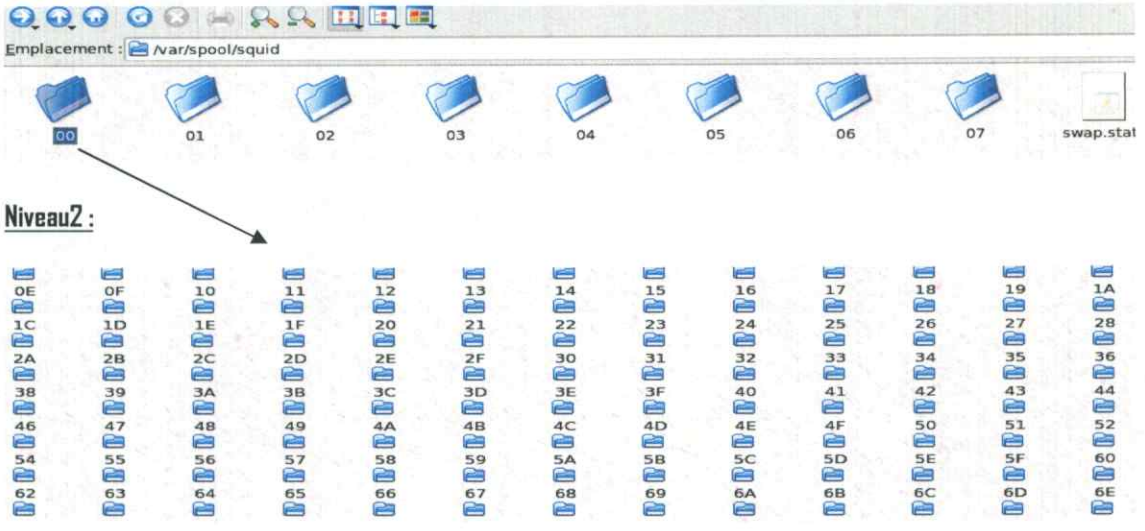


-Figure 4.23-

Test :

Le cache de squid se trouve dans /var/spool/squid.

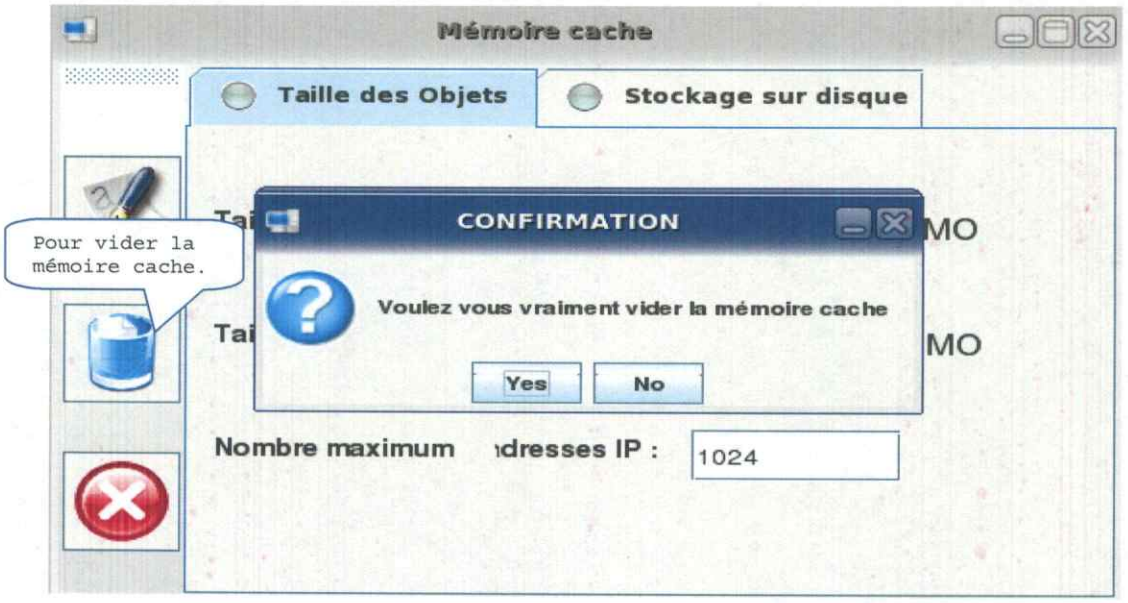
Niveau 1 :



-Figure 4.24-

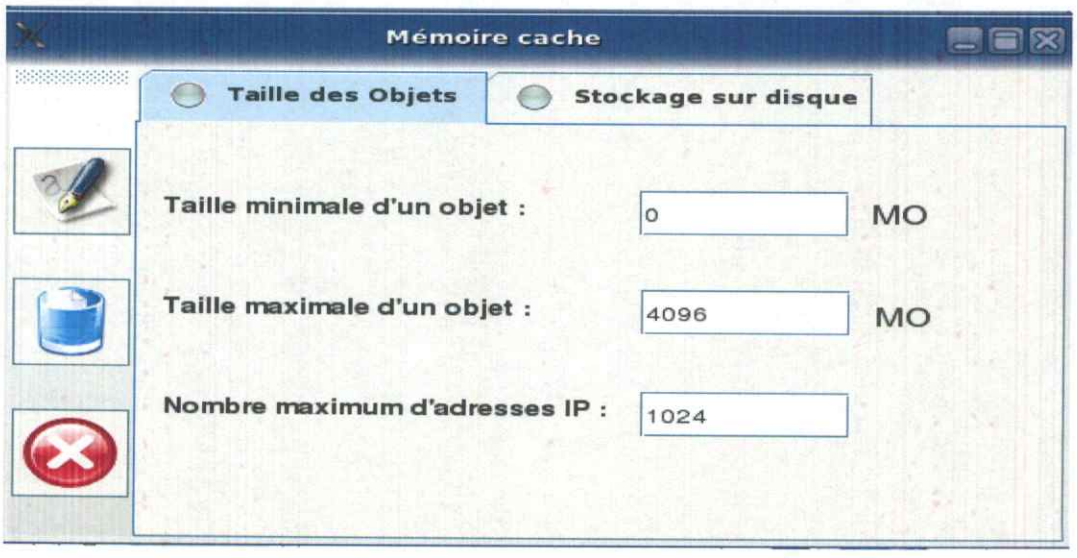
- Suppression des éléments de mémoire cache :

Il est possible de vider les éléments de la mémoire cache :



-Figure 4.25-

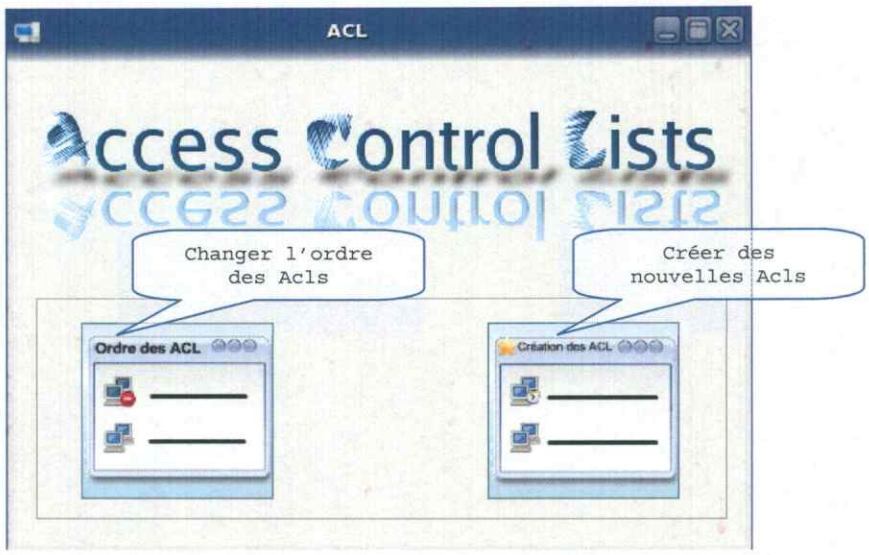
- Pour limiter la taille des objets ainsi que le nombre d'adresse IP stockés dans la mémoire cache :



-Figure 4.26-

- Interface de configuration des acls :

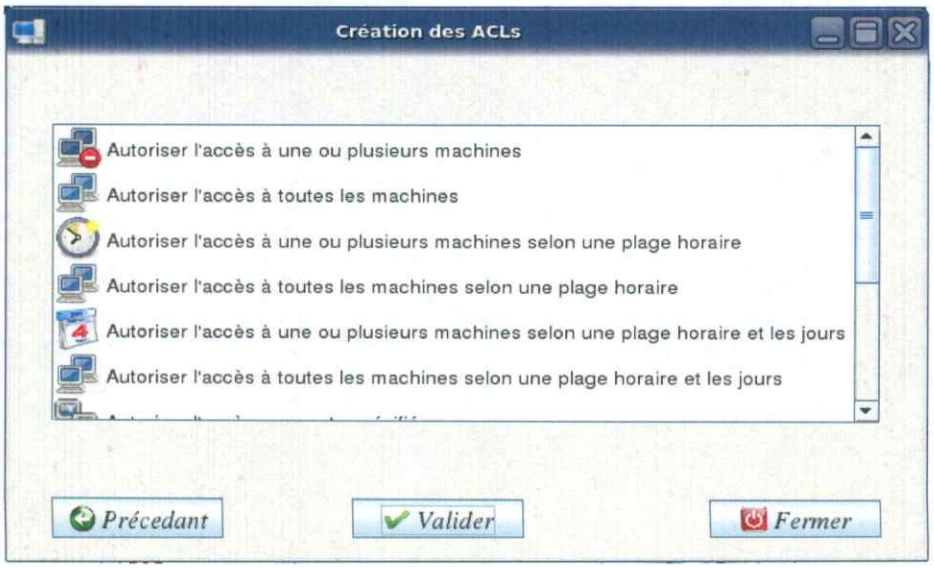
Notre outil permet de gérer les ACLs, ainsi de les visualiser et modifier leur ordre.



-Figure 4.27-

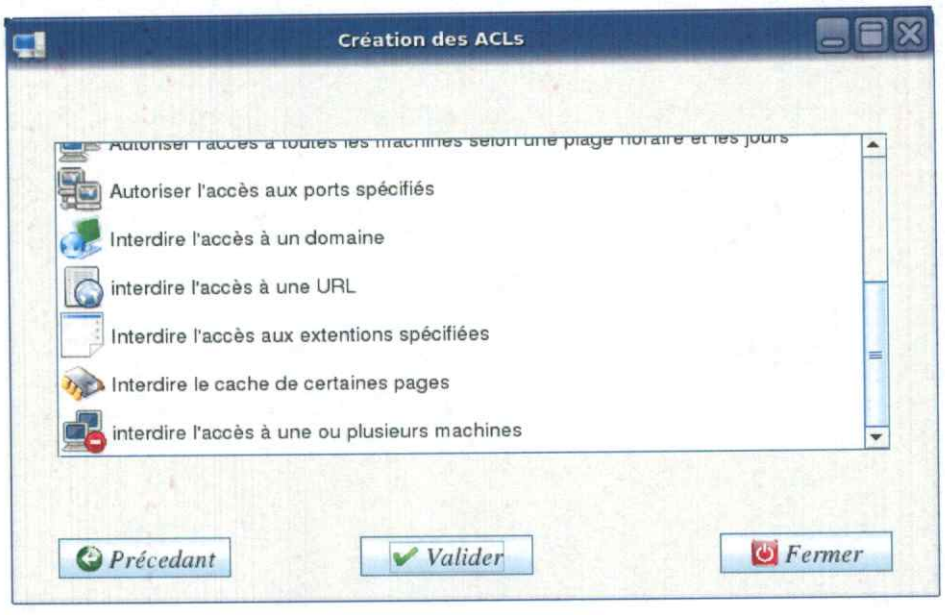
- Interface de la gestion des ACLs :

Il existe plusieurs types d'ACLs qui sont regroupés dans une liste, l'utilisateur n'a qu'à sélectionner le type de l'ACL qu'il veut gérer. Ci-dessous la liste des types des ACLs :

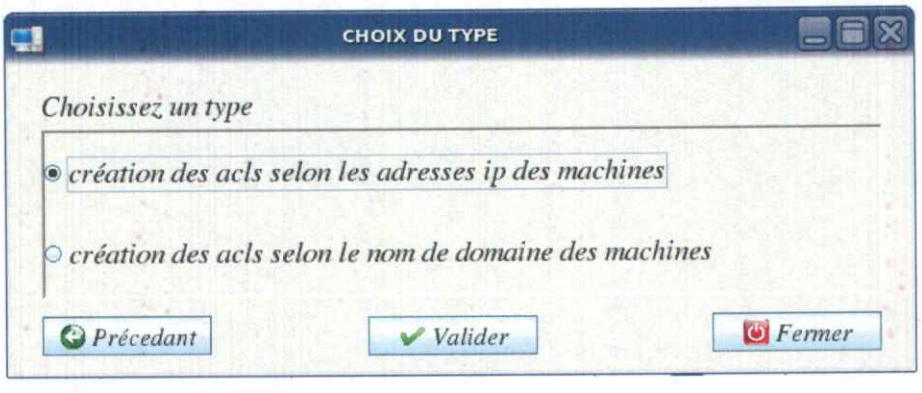


-Figure 4.28-

La suite de la liste :



- Boite de dialogue du choix de type des ACLs qui autorisent l'accès à une ou plusieurs machines :



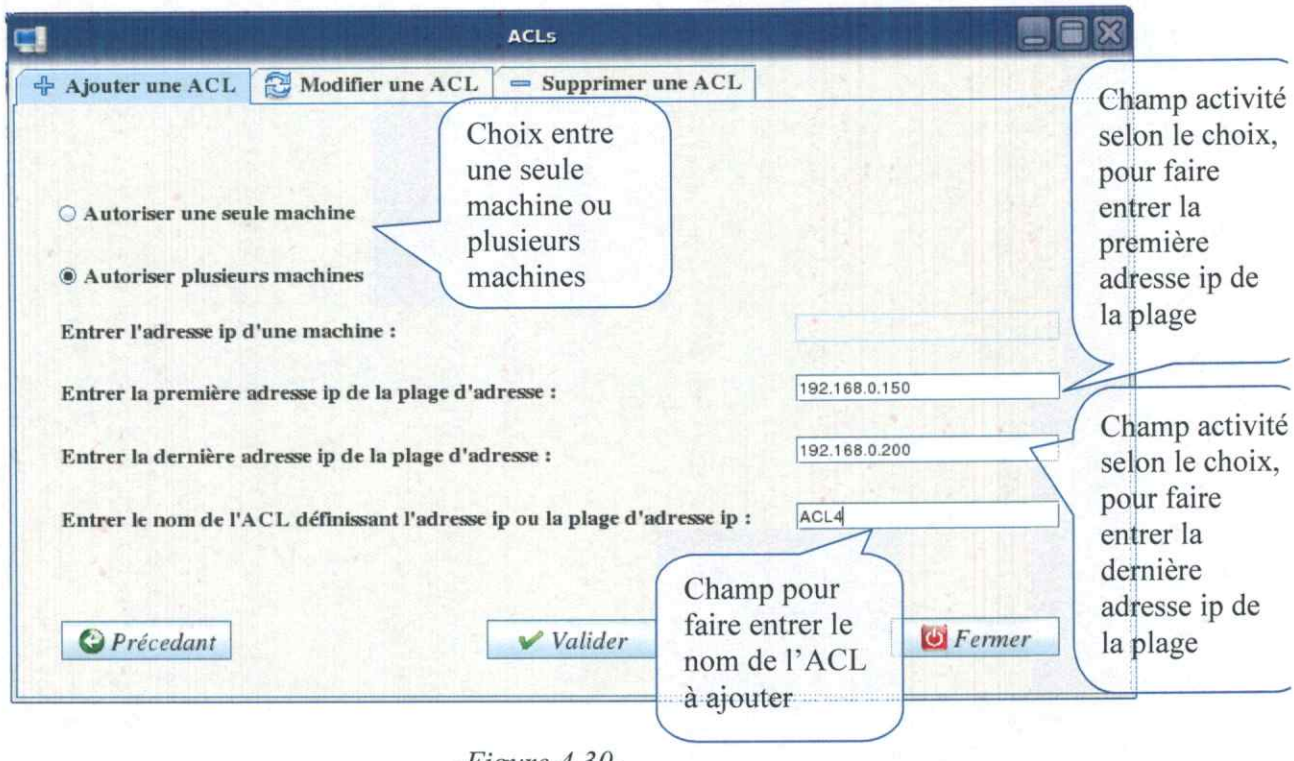
-Figure 4.29-

- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon les adresses ip des machines :

L'utilisateur peut ajouter, modifier, ou supprimer des ACLs de ce type

1) l'interface de l'ajout :

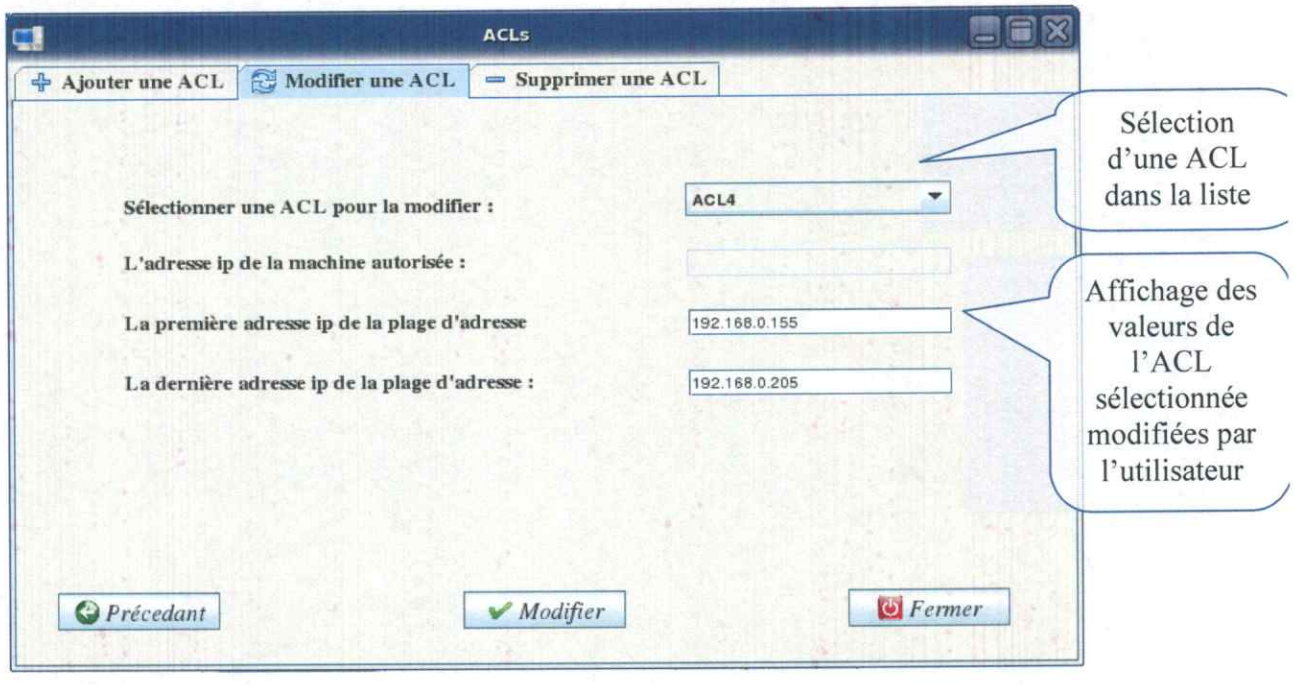
L'utilisateur peut ajouter des ACLs en sélectionnant le choix entre une seule machine ou plusieurs machines, selon son choix les champs des adresses ip seront activés pour les utiliser, puis il fait entrer les adresses IP dans les champs activés, ensuite il fait entrer le nom de l'ACL, et il valide.



-Figure 4.30-

2) l'interface de la modification :

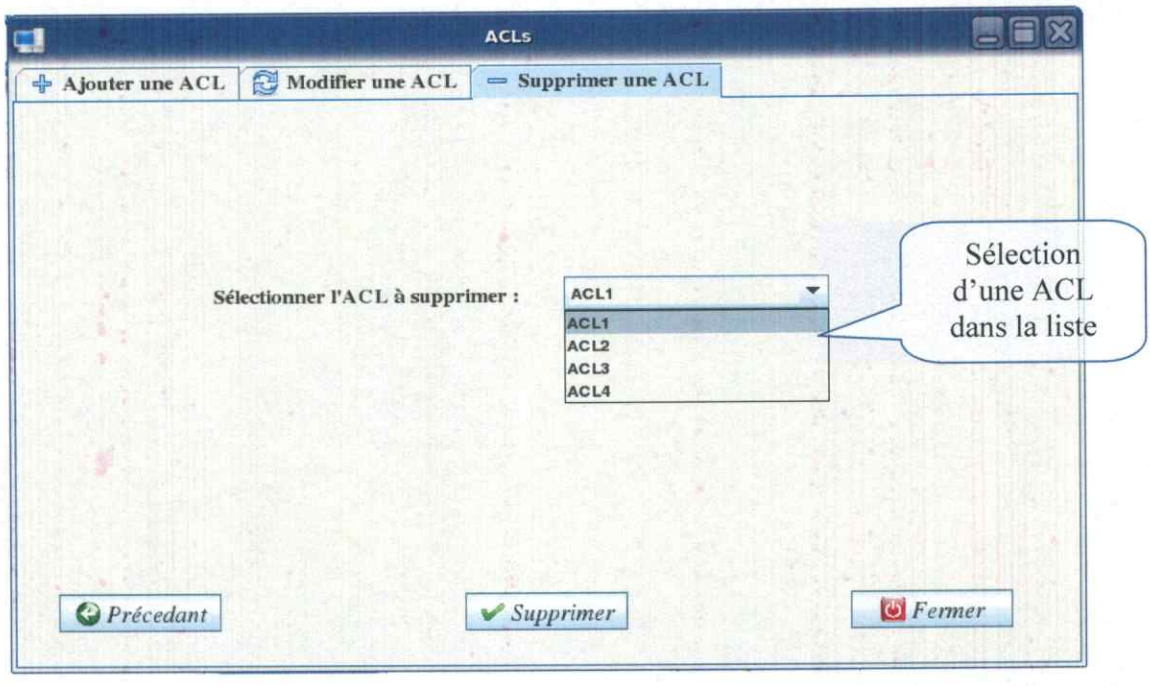
Pour modifier une ACL, notre outil affiche une liste des ACLs ajoutées dans ce type, puis l'utilisateur sélectionne l'ACL qu'il veut modifier,ensuite le système lui affiche les valeurs des paramètres de l'ACL sélectionnée, donc l'utilisateur n'a qu'à modifier les valeurs , et valider.



-Figure 4.31-

3) l'interface de la suppression :

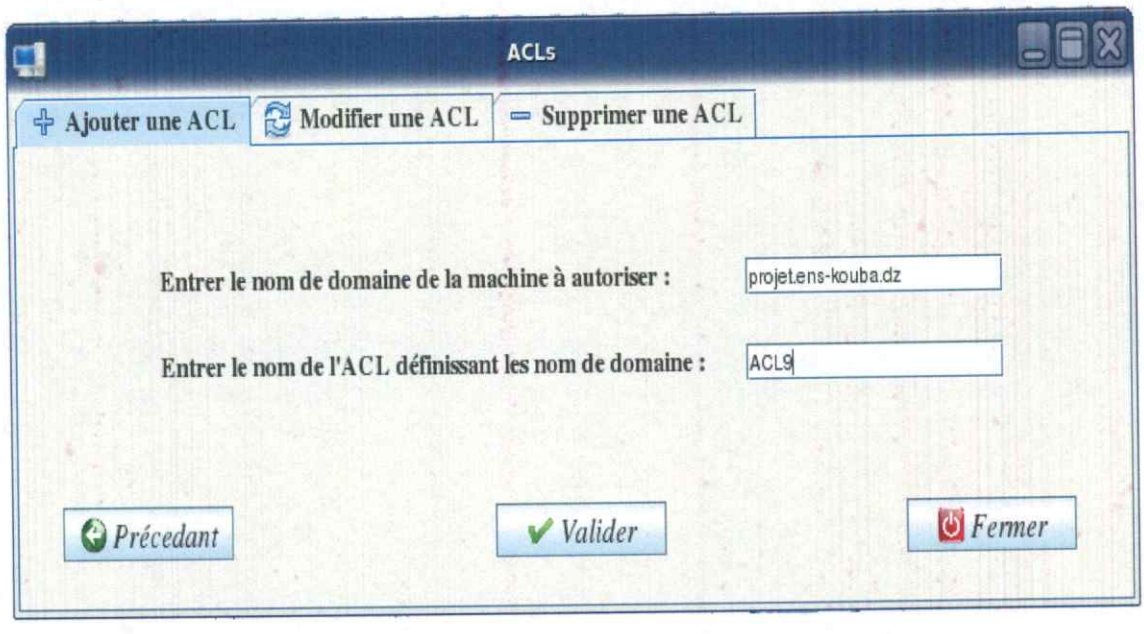
Pour supprimer une ACL, notre outil affiche la liste de toutes les ACLs ajoutées dans ce type, l'utilisateur sélectionne une ACL, puis il l'a supprime.



-Figure 4.32-

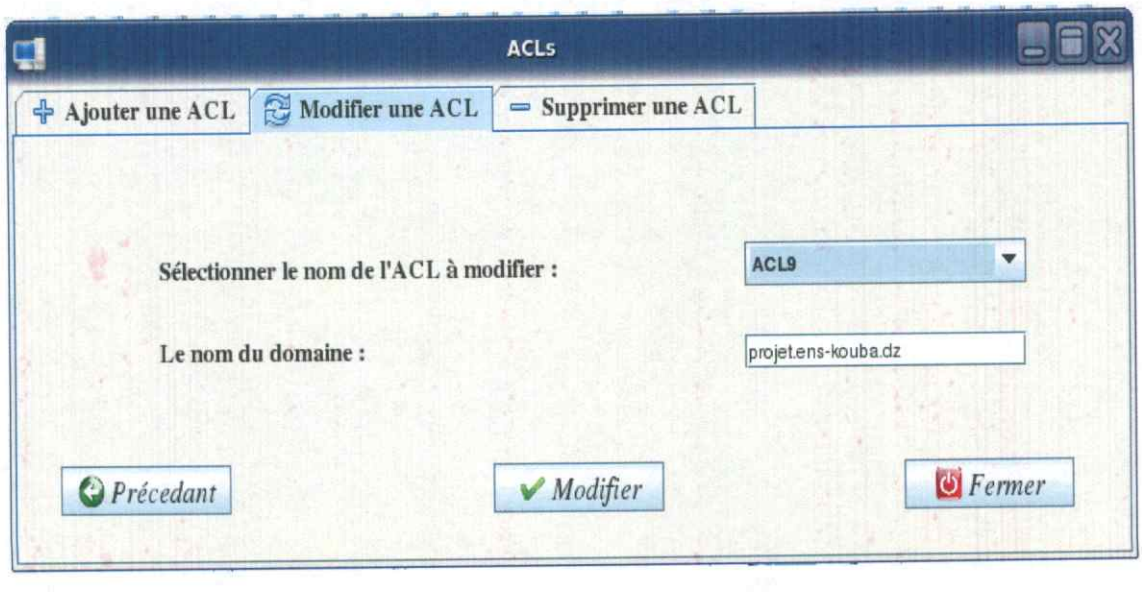
- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon les noms des domaines des machines :

1) Interface de l'ajout :



-Figure 4.33-

2) Interface de la modification :

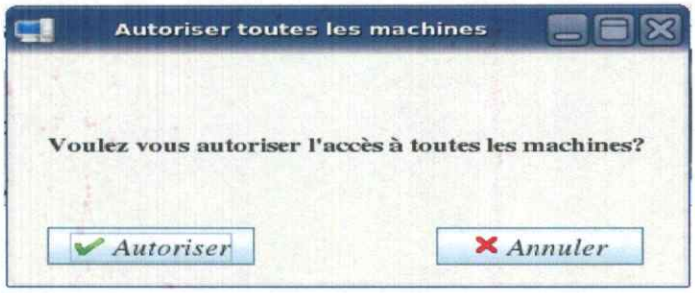


-Figure 4.34-

3) Interface de la suppression : voir -*Figure 4.32*-

- Interface de l'ACL qui autorise l'accès à toutes les machines :

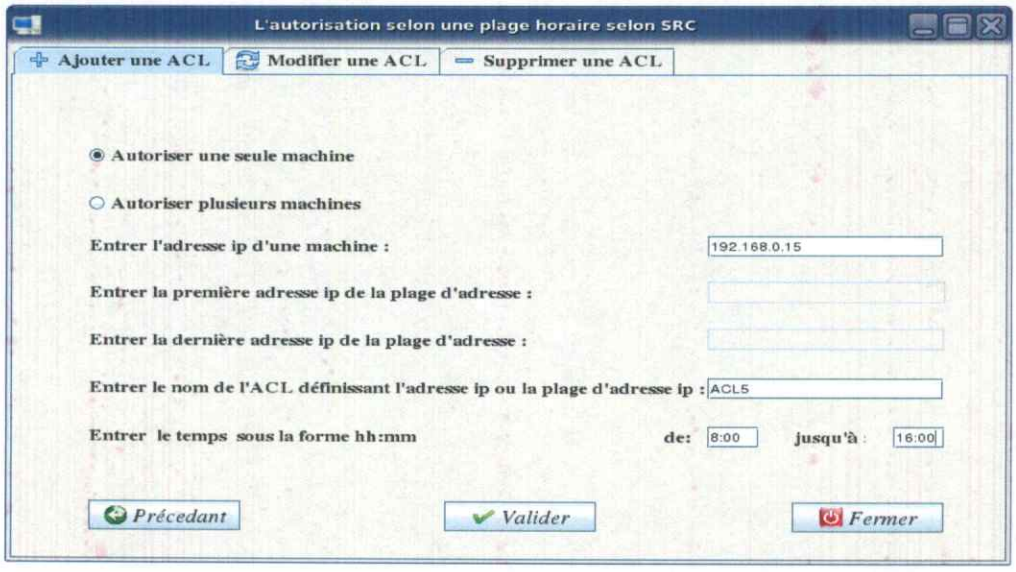
En utilisant cette ACL, l'utilisateur peut donner l'accès à toutes les machines, comme il peut l'annuler.



-*Figure 4.35*-

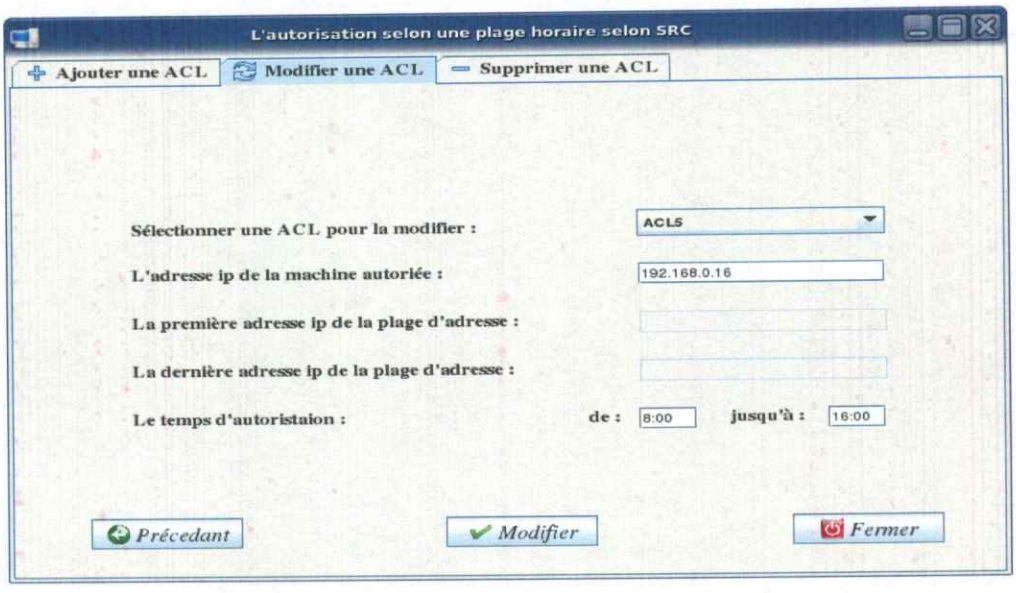
- Interface du choix de type des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire: voir -*Figure 4.29*-
- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire, en utilisant les adresses ip des machines :

1) Interface de l'ajout :



-*Figure 4.36*-

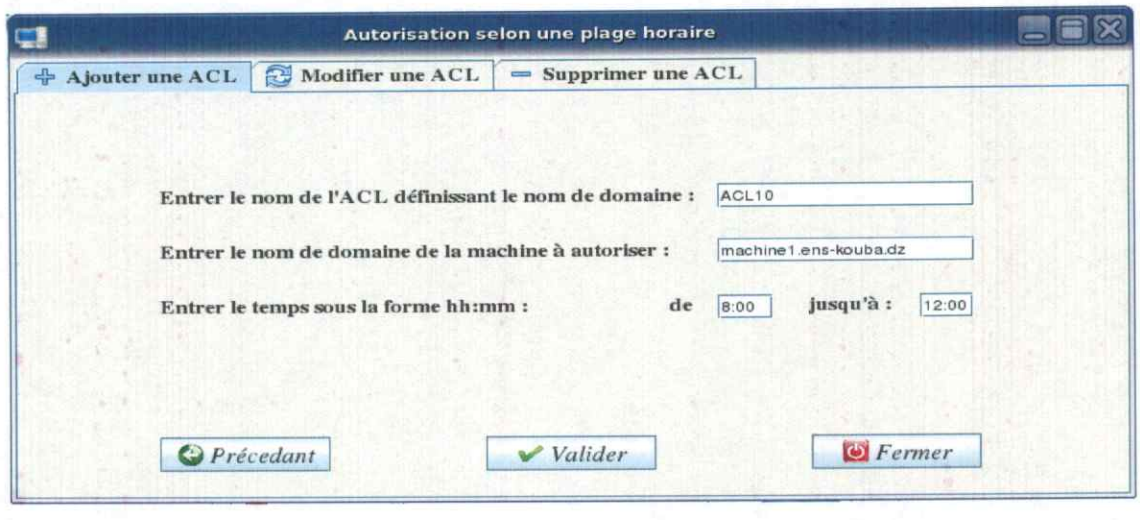
- Interface de la modification :



-Figure 4.37-

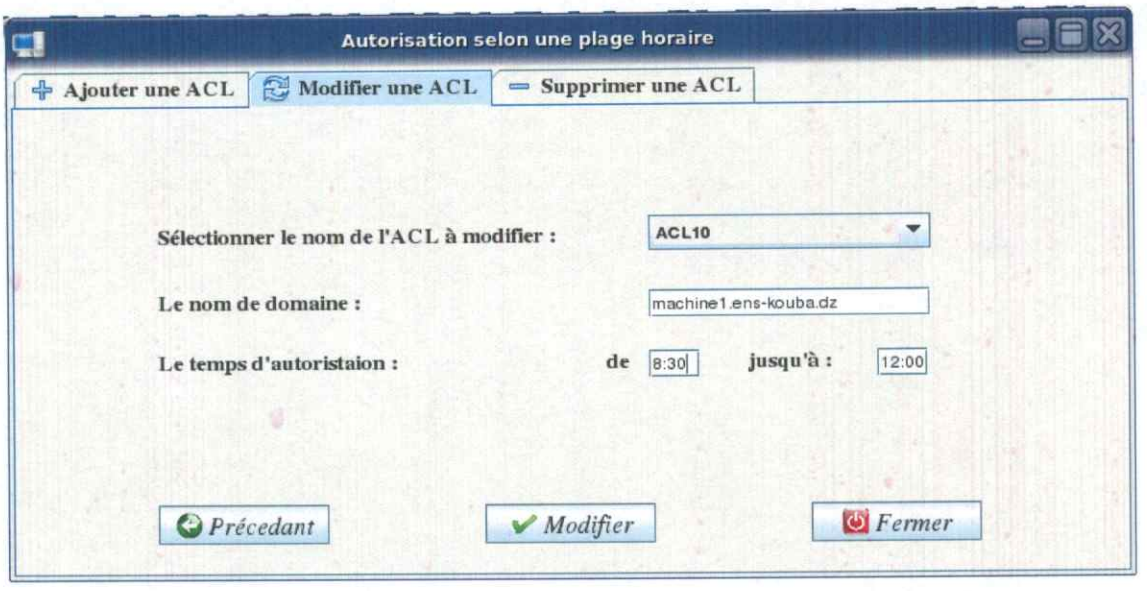
- Interface de la suppression : voir *-Figure 4.32-*
- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire, en utilisant les noms de domaines des machines :

1) Interface de l'ajout :



-Figure 4.38-

2) Interface de la modification :

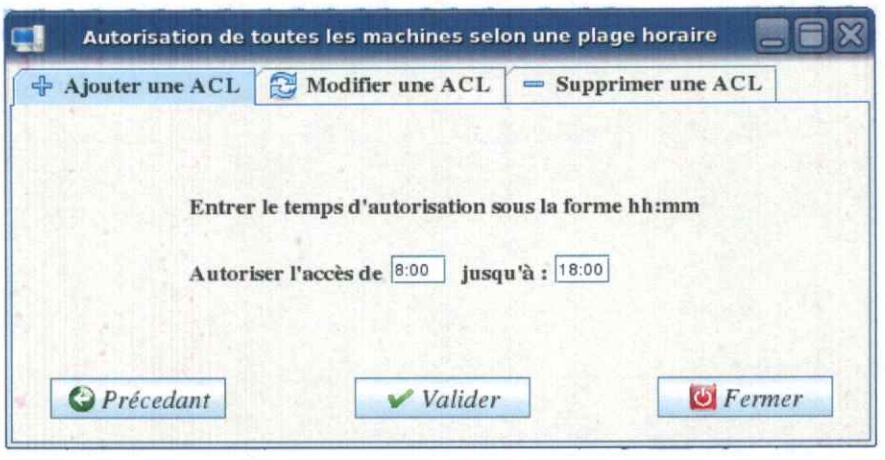


-Figure 4.39-

3) Interface de la suppression : voir *-Figure 4.32-*

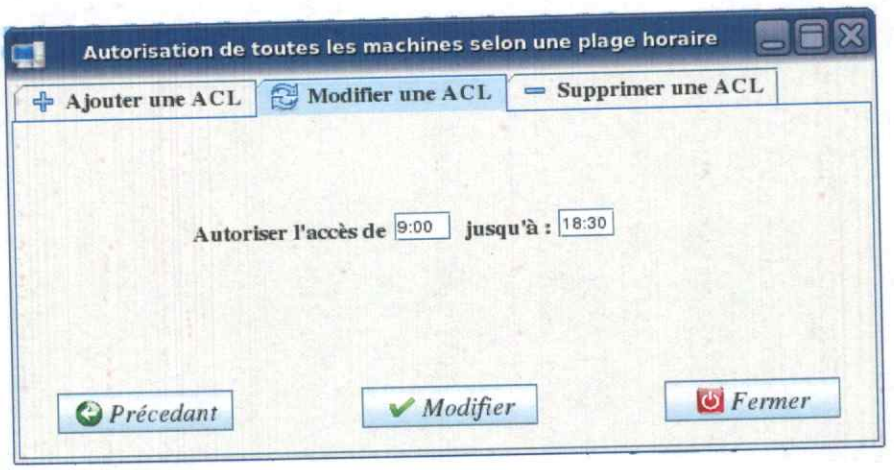
- Interface de l'ACL qui autorise l'accès à toutes les machines selon une plage horaire :

1) Interface de l'ajout :



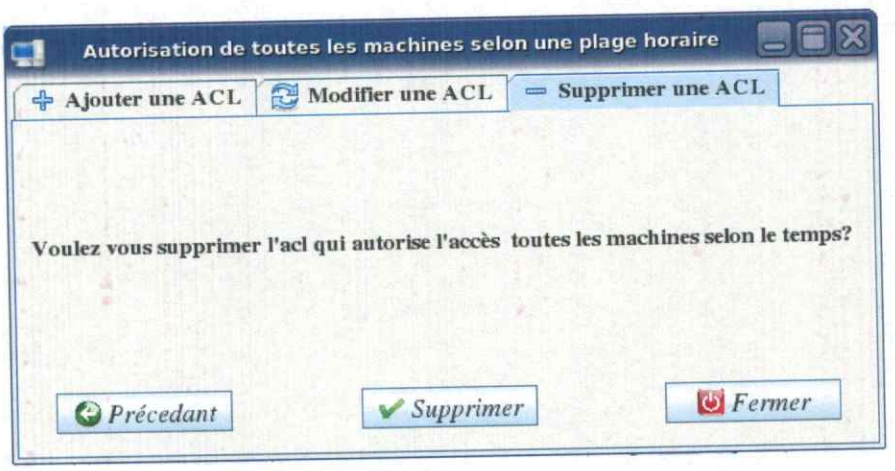
-Figure 4.40 -

2) Interface de la modification :



-Figure 4.41-

3) Interface de la suppression :

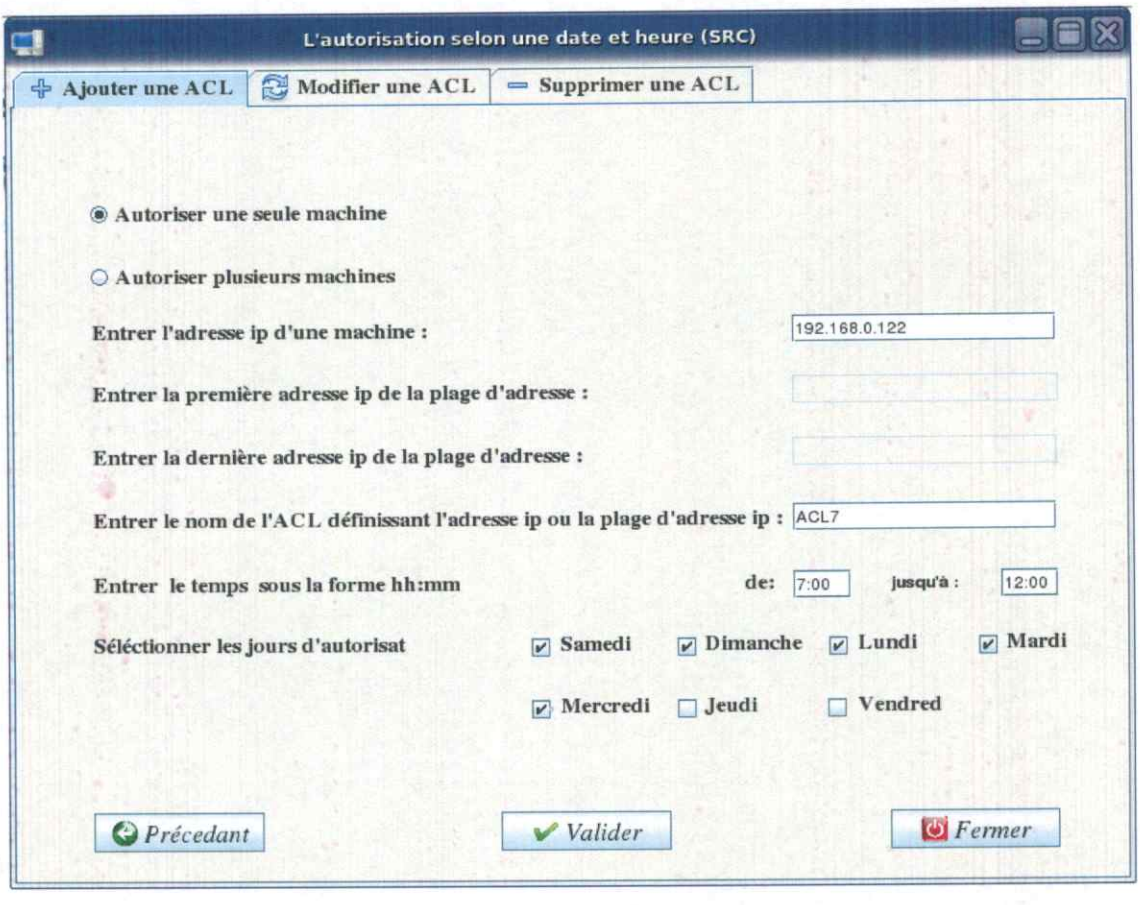


-Figure 4.42-

- Interface du choix de type des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire et les jours : voir *-Figure 4.29-*

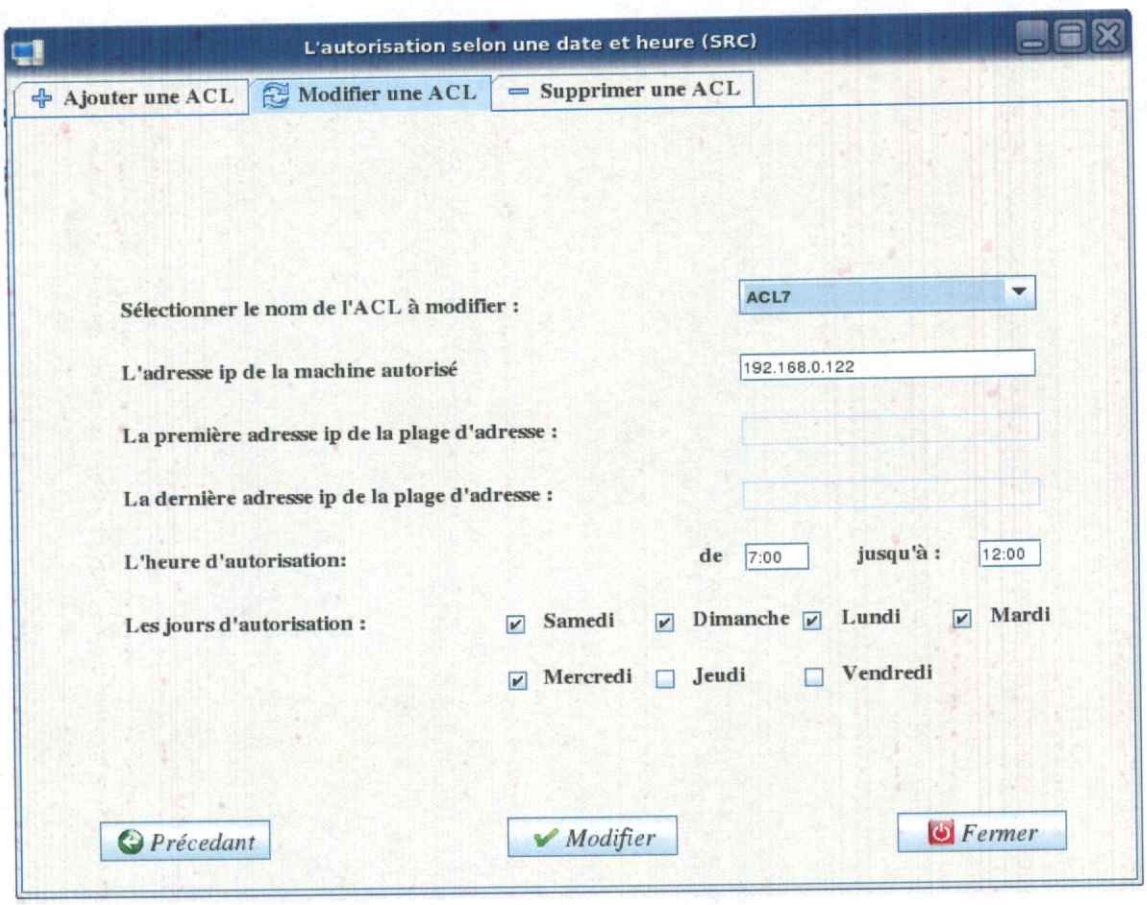
- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire et les jours, en utilisant les adresses ip des machines :

1) Interface de l'ajout :



-Figure 4.43-

2) Interface de la modification :

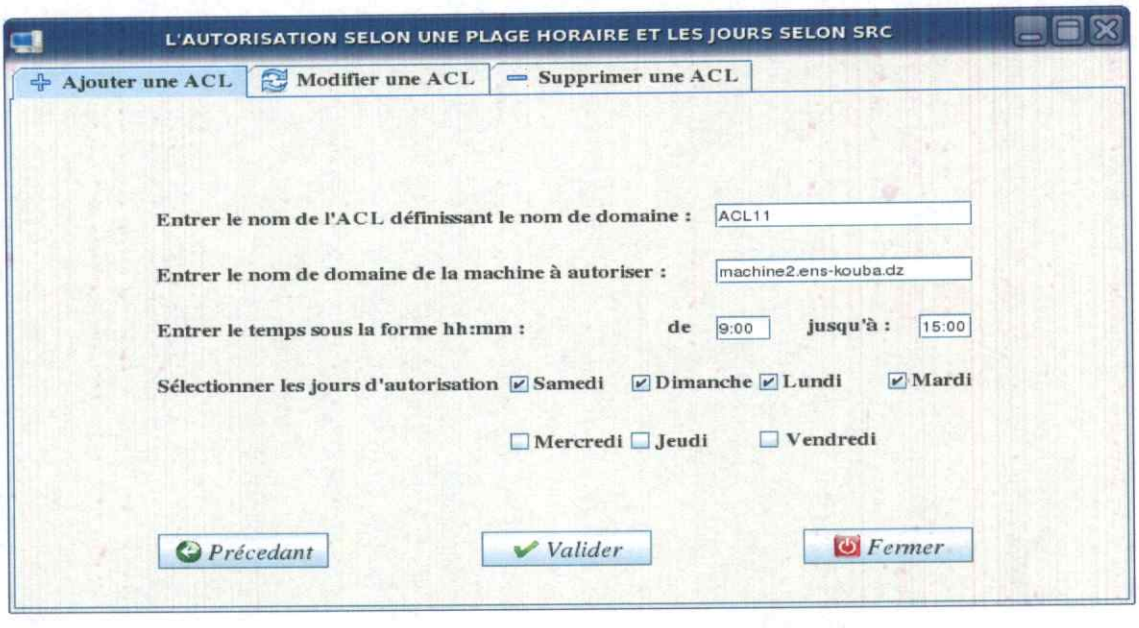


-Figure 4.44-

3) Interface de la suppression : voir *-Figure 4.32-*

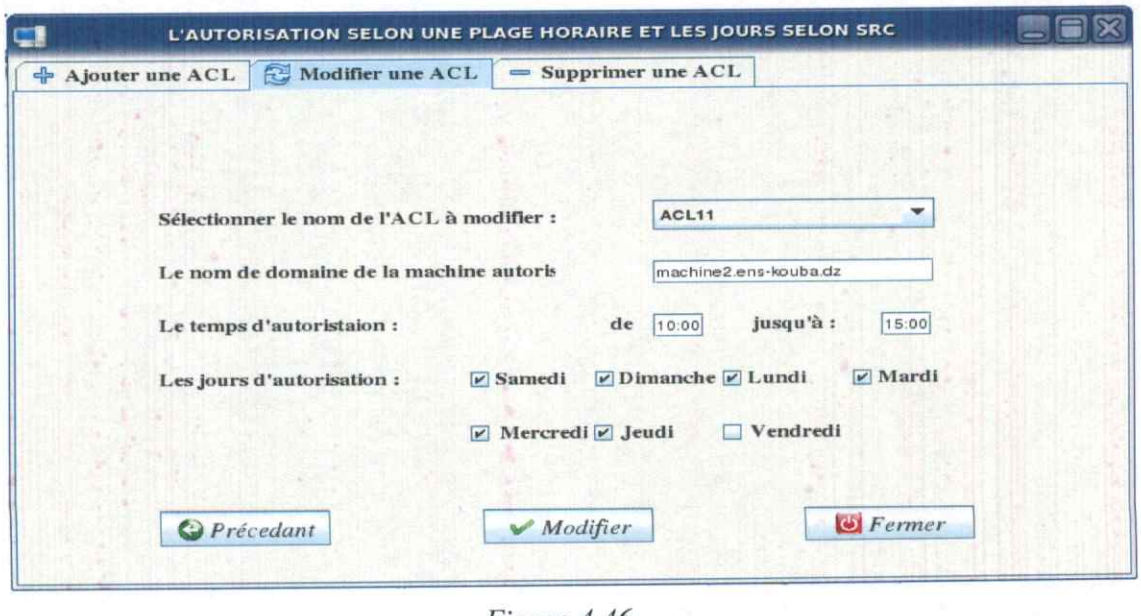
- Interface des ACLs qui autorisent l'accès à une ou plusieurs machines selon une plage horaire et les jours, en utilisant les noms de domaines des machines :

1) Interface de l'ajout :



-Figure 4.45-

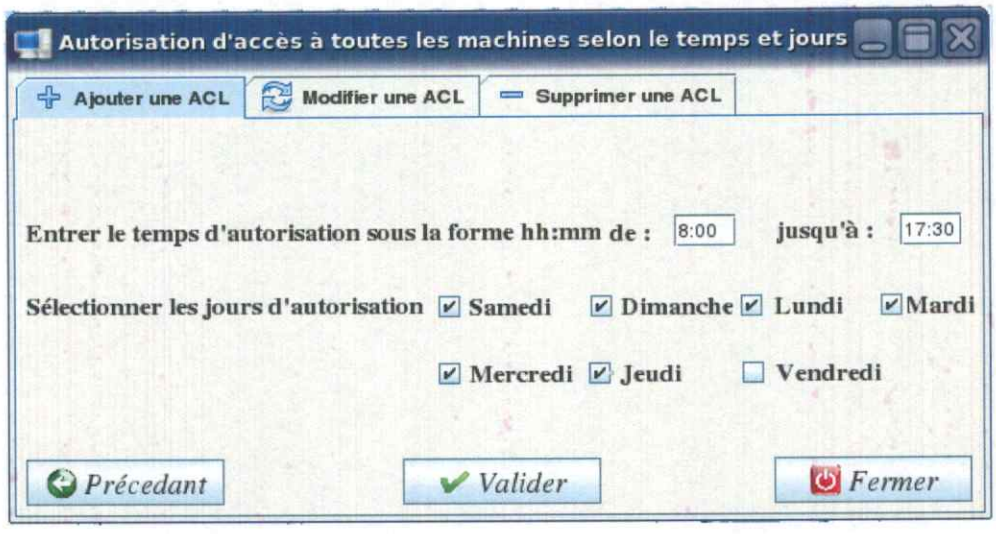
2) Interface de la modification :



-Figure 4.46-

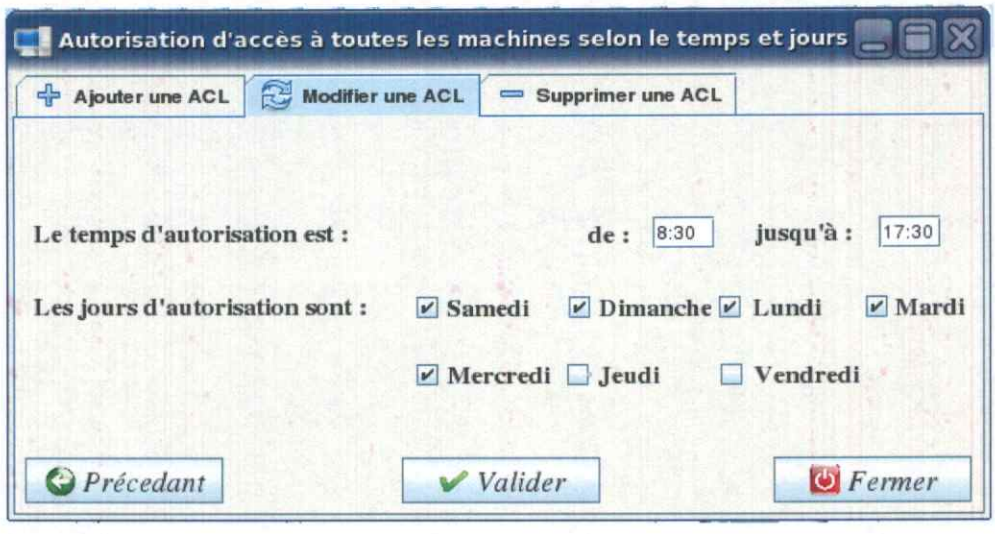
- Interface de la suppression : voir *-Figure 4.32-*
- Interface de l'ACL qui autorise l'accès à toutes les machines selon une plage horaire et les jours :

1) Interface de l'ajout :



-Figure 4.47-

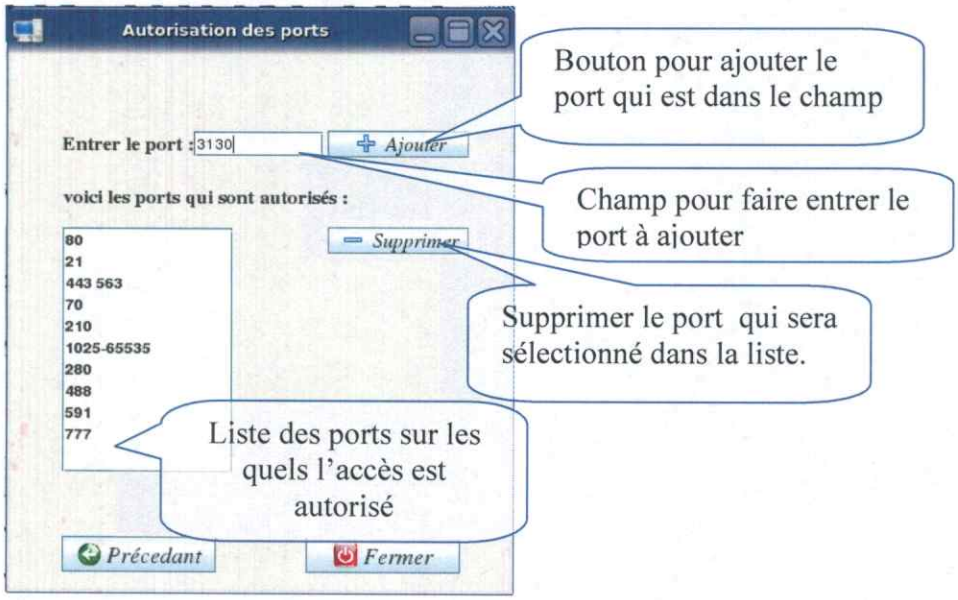
2) Interface de la modification :



-Figure 4.48-

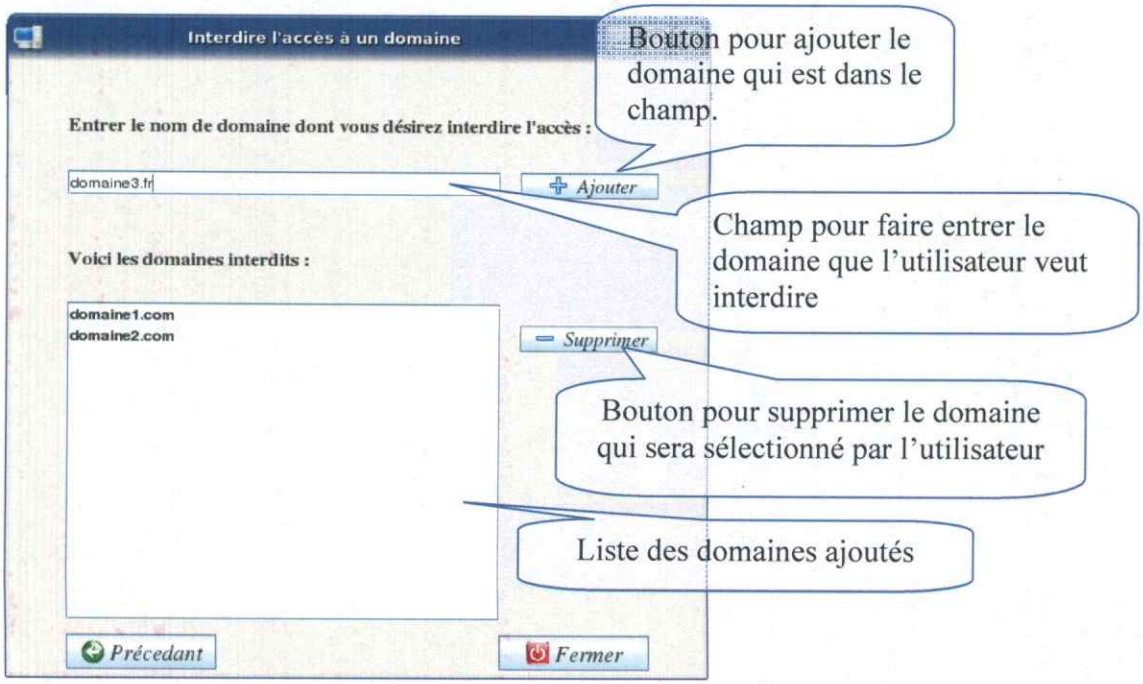
3) Interface de la suppression : voir -*Figure 4.42*-

- Interface de l'autorisation de l'accès sur des ports spécifiés : L'utilisateur peut ajouter/ supprimer des port sur les quels il autorise l'accès.



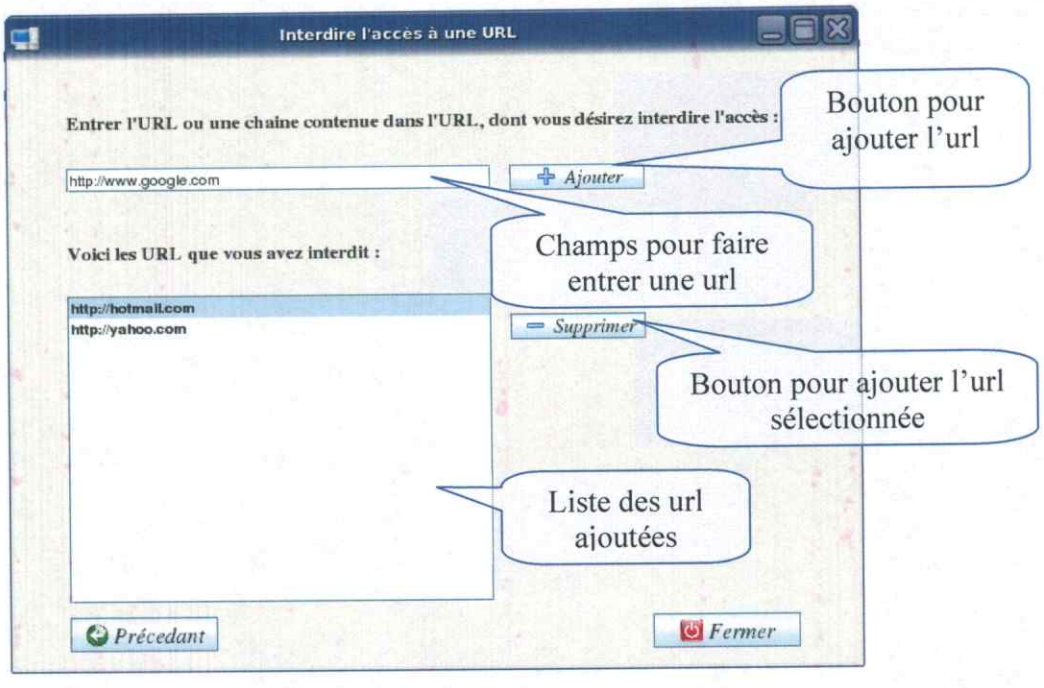
-Figure 4.49-

- Interface de l'ACL qui interdit l'accès à des domaines : L'utilisateur peut interdire l'accès à des domaines.



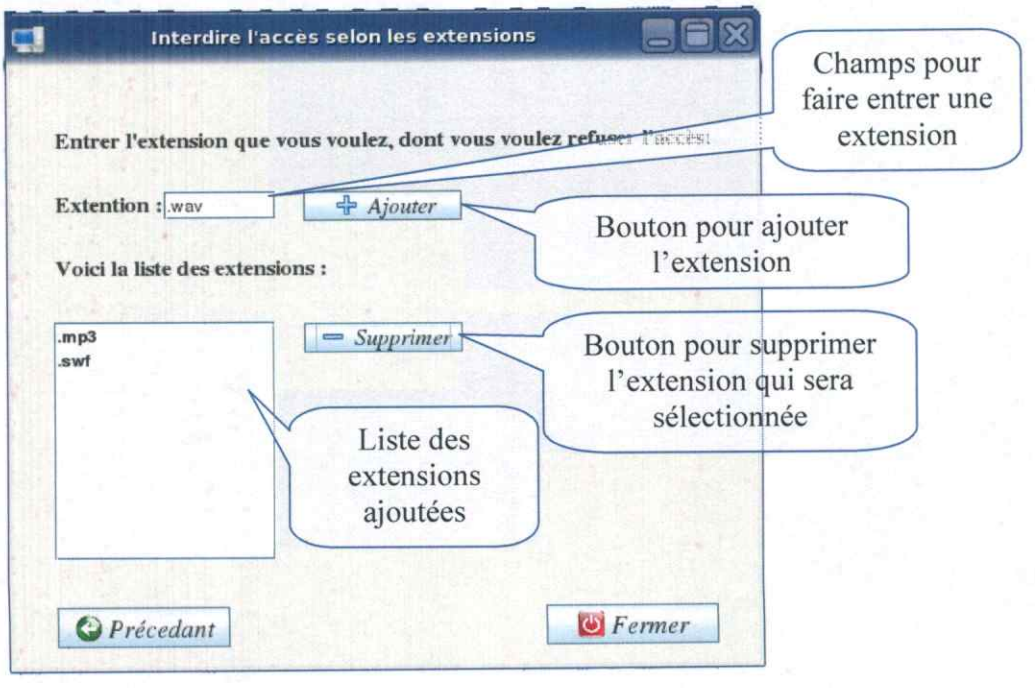
-Figure 4.50-

- Interface de l'ACL qui interdit l'accès à des URLs :



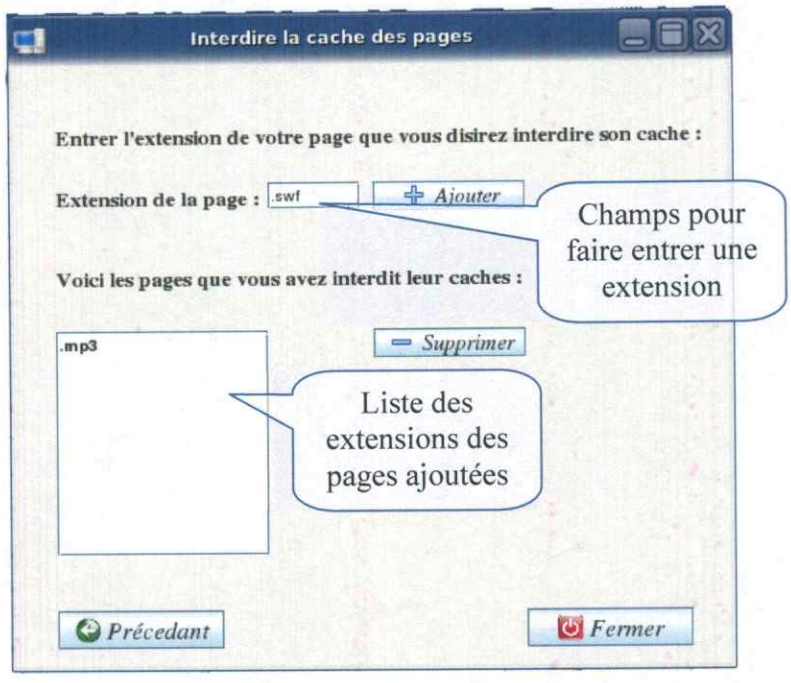
-Figure 4.51-

- Interface des ACLs qui interdisent l'accès à des extensions spécifiques :



-Figure 4.52-

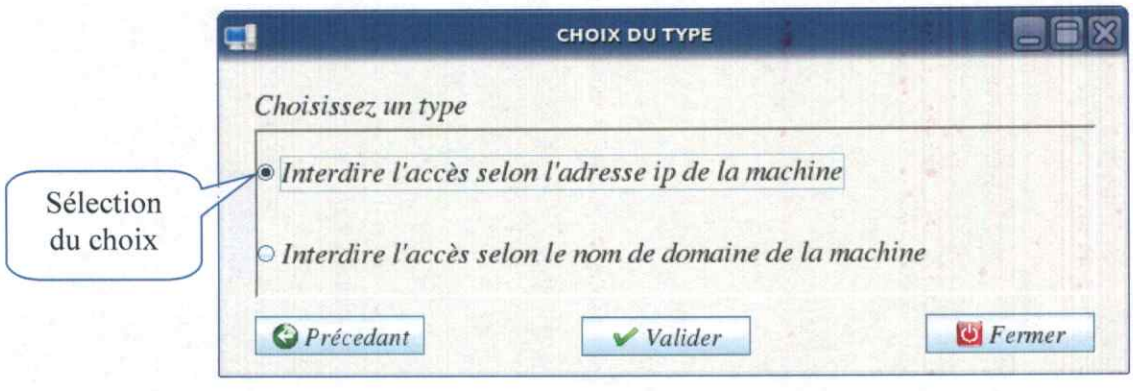
- Interface de l'interdiction du cache de certaine page :



-Figure 4.53-

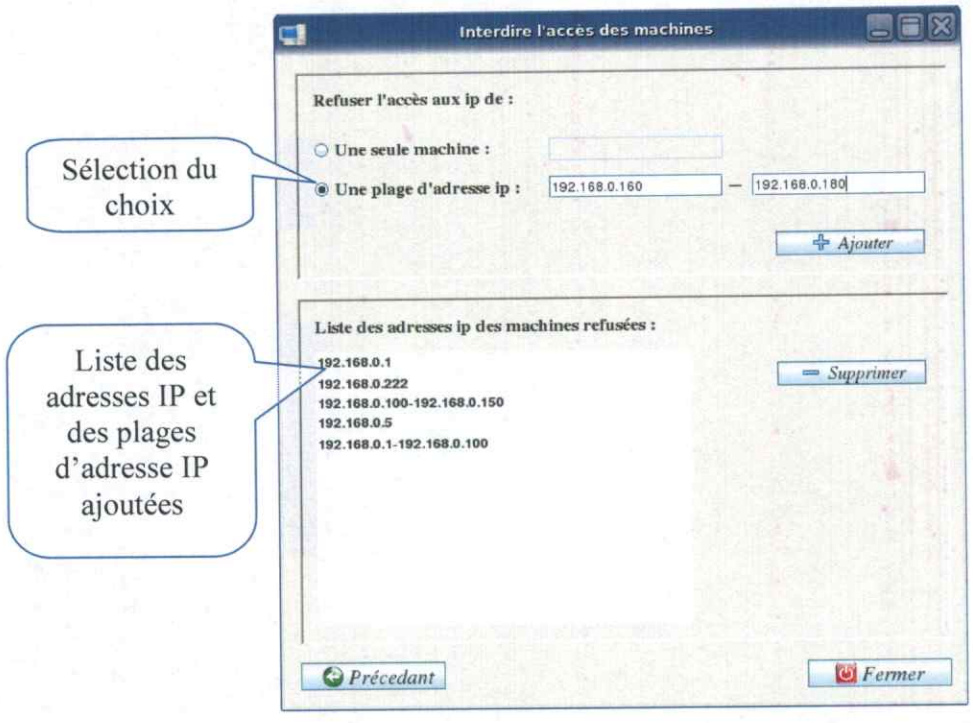
- Interface du choix de l'ACL qui interdit l'accès à une ou plusieurs machines :

L'utilisateur peut interdire l'accès à une ou plusieurs machines selon leurs noms de domaine ou leurs adresses IP.



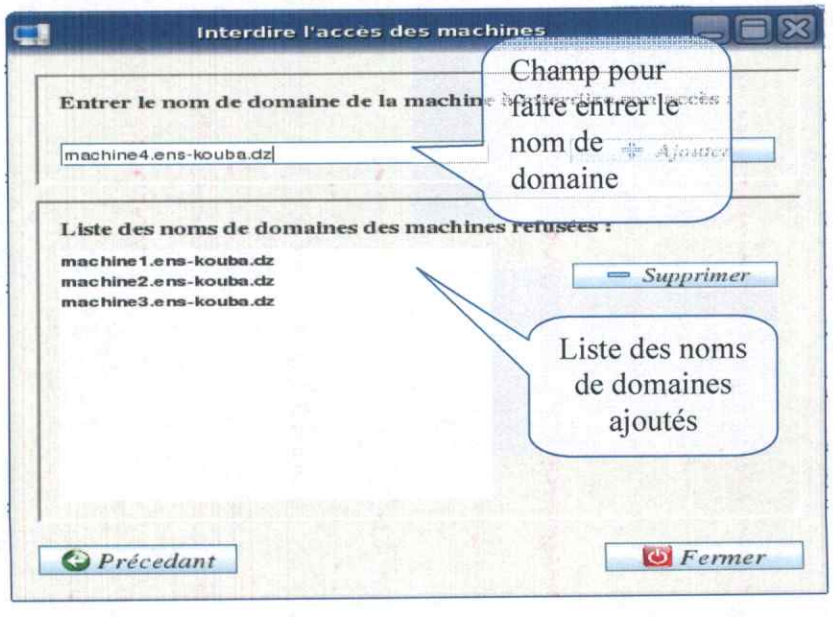
-Figure 4.54-

- Interface de l'ACL qui interdit l'accès selon l'adresse ip des machines: L'utilisateur peut interdire l'accès à une seule machine ou à une plage d'adresse IP.



-Figure 4.55-

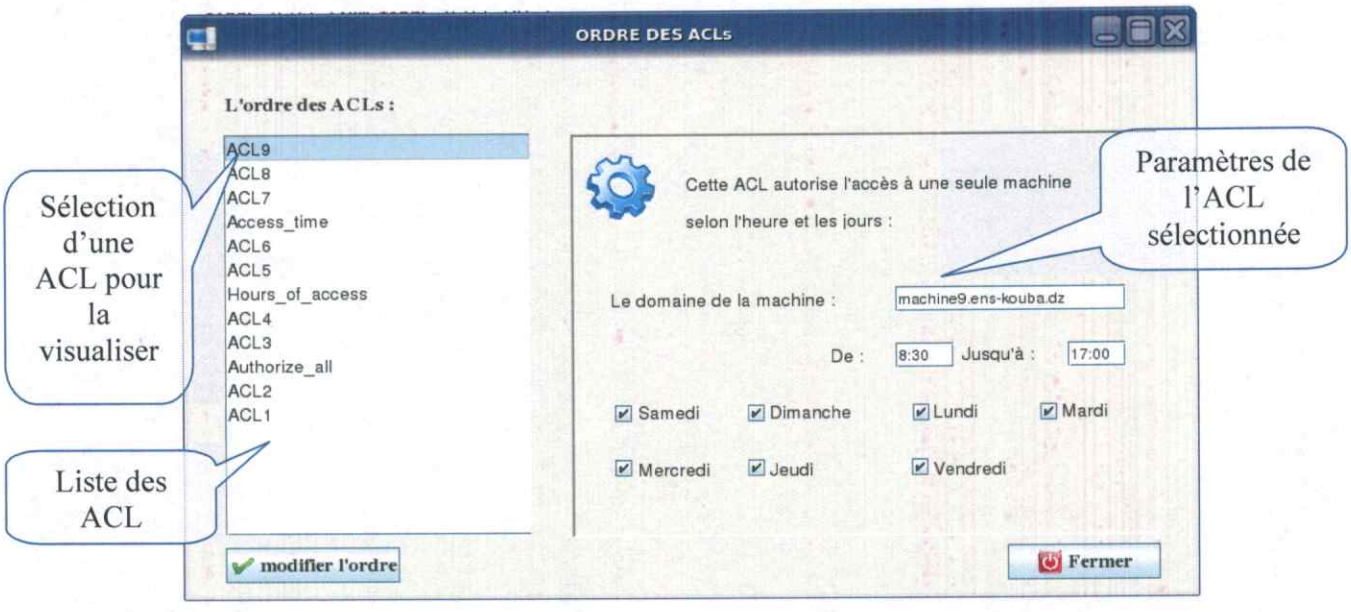
- Interface de l'ACL qui interdit l'accès selon le nom de domaine de la machine :



-Figure 4.56

- Interface de l'ordre des ACLs :

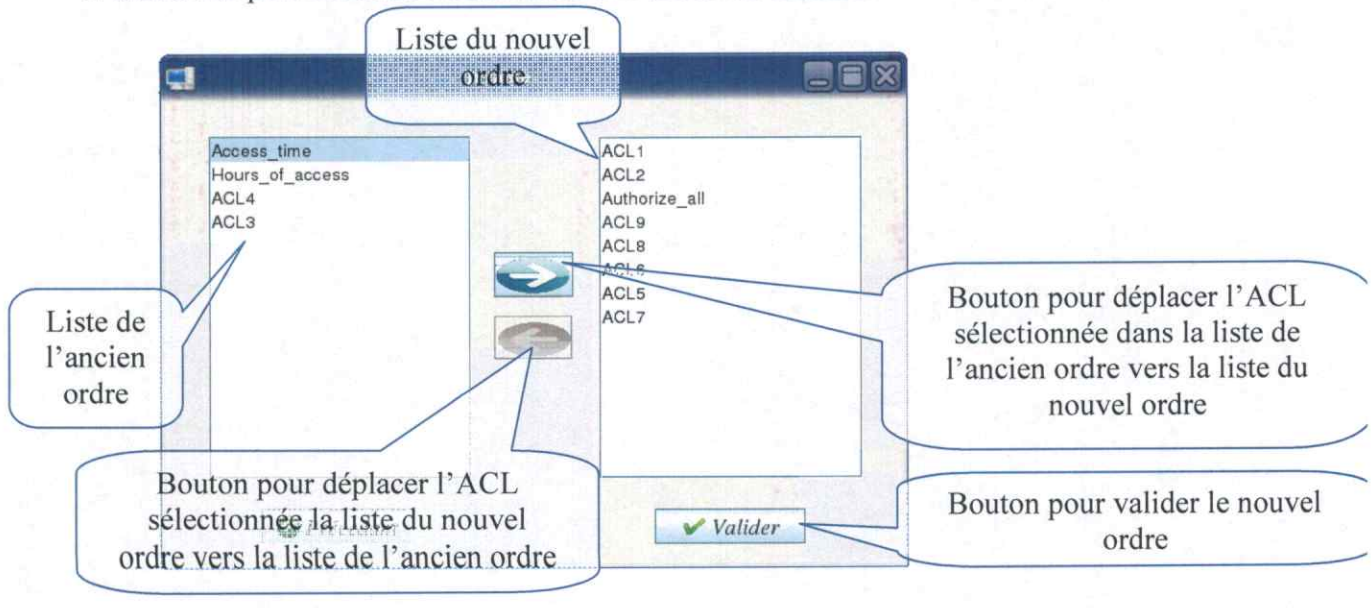
L'utilisateur peut voir l'ordre des ACL, comme il peut visualiser chaque ACL en la sélectionnant :



-Figure 4.57-

- Interface de la modification de l'ordre des ACLs :

L'utilisateur peut modifier l'ordre des ACLs selon ses besoins.



-Figure 4.58

Interface de configuration de SquidGuard :

-Figure 4.59-

Le Menu :

Il se compose de 4 items (numéroté de 1 à 4 dans la figure 5.2)

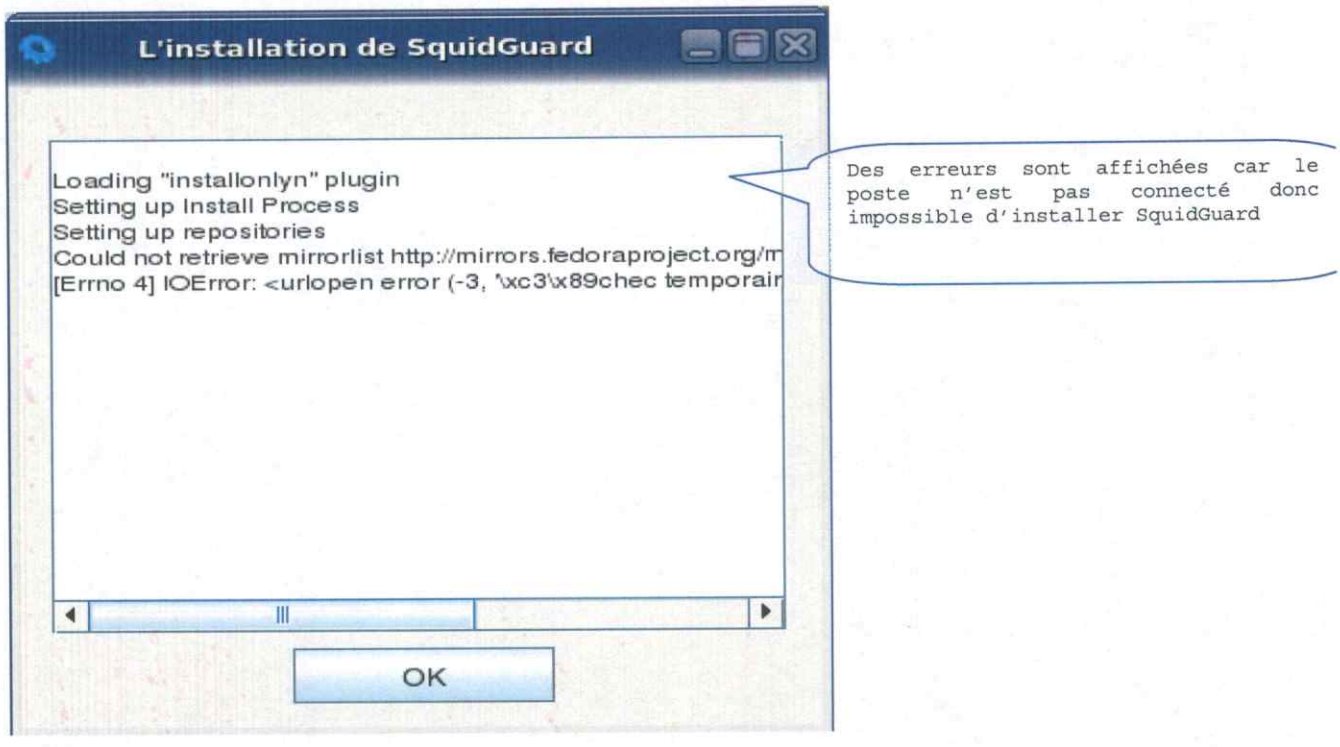
- L'item **N°=1** : pour revenir à la page d'accueil.
- L'item **N°=2** : pour revenir à l'interface de configuration de SquidGuard.
- L'item **N°=3** : pour installer SquidGuard.
- L'item **N°=4** : pour compiler les sources.

- L'installation de SquidGuard :

Nécessite la connexion, une boite de dialogue permet de visualiser les étapes d'installation :

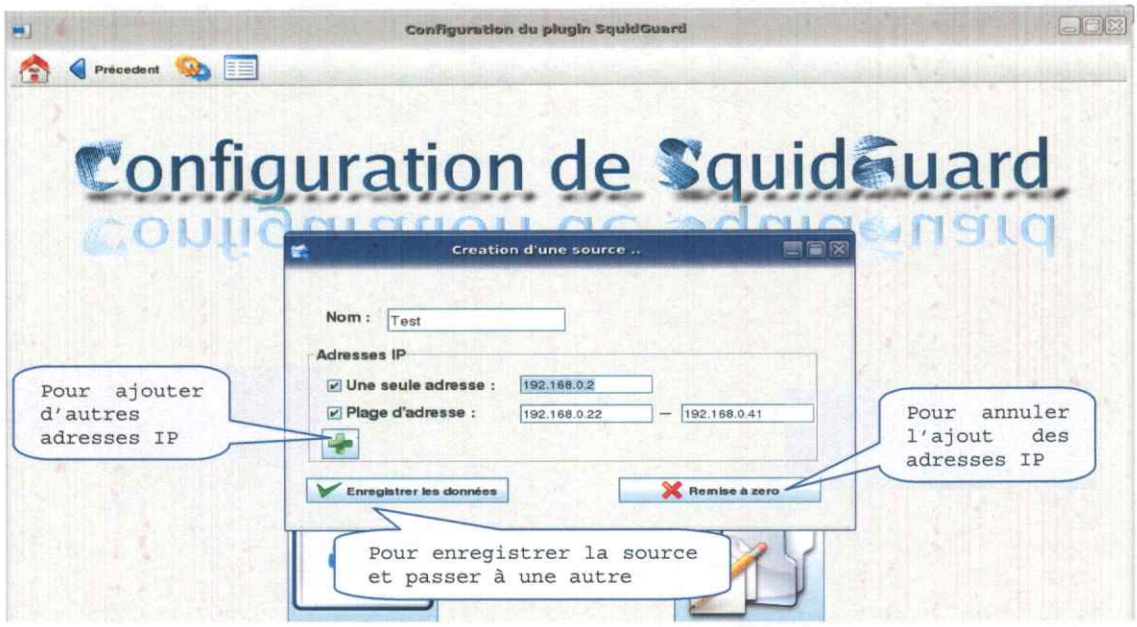
Test :

L'installation a échoué car le pc n'est pas relié à INTERNET.



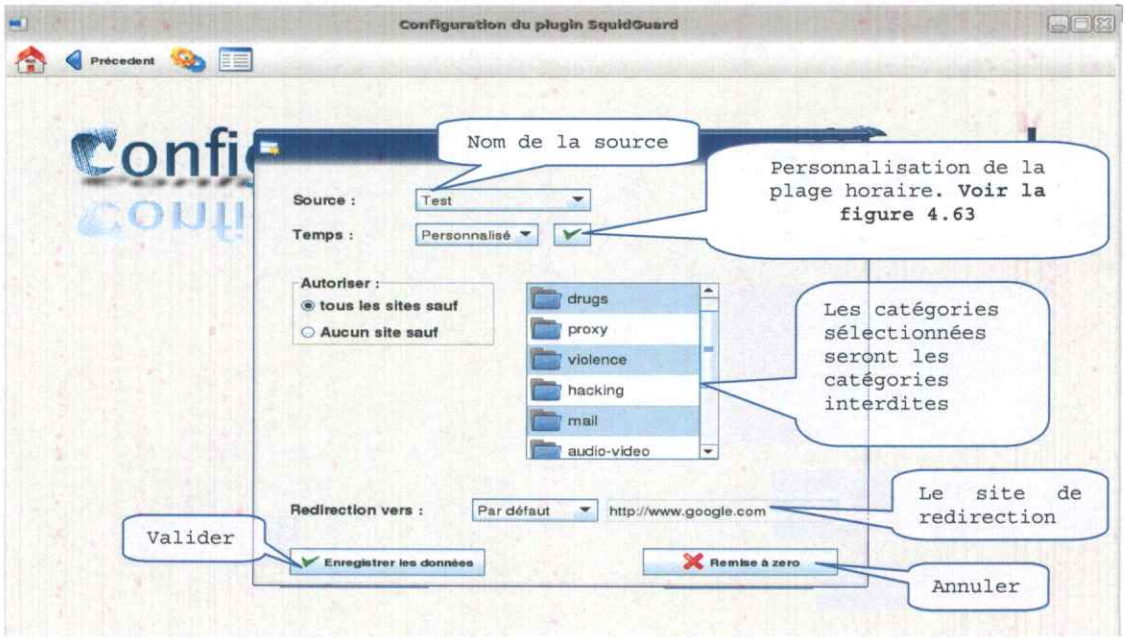
-Figure 4.60-

- Création des sources :



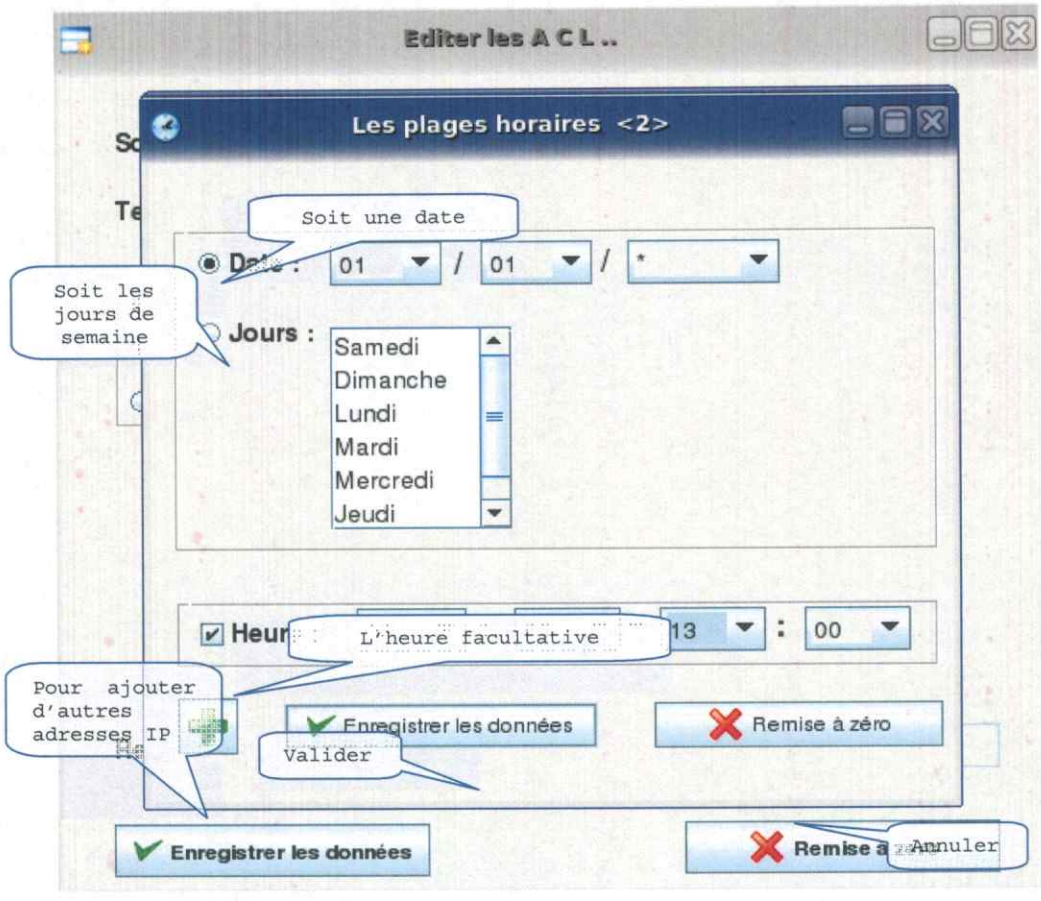
-Figure 4.61-

- Interface de création du fichier de configuration de SquidGuard :



-Figure 4.62-

- Personnalisation de la plage horaire :



-Figure 4.63-

Lorsque toutes les sources sont entrées le fichier squidGuard.conf est généré :

-Figure 4.64-

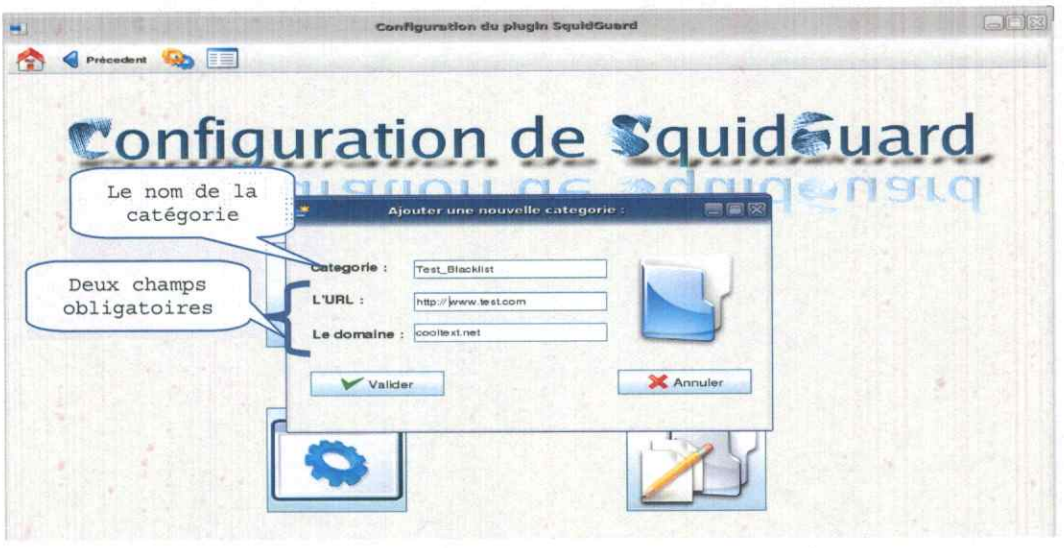
Test :

Le fichier SquidGuard.conf généré pour l'exemple précédent est :

```
#
# CONFIG FILE FOR SQUIDGUARD#
# See http://www.squidguard.org/config/ for more examples
#

dbhome /var/squidGuard/blacklists/blacklists
logdir /var/log/squidGuard



time Test_time {
date  *.01.01 03:00-13:00
weekly  saturdays sundays tuesdays
}


src Test {
ip 192.168.0.22-192.168.0.41
ip 192.168.0.2
}

dest drugs{
log  drugs
domainlist  drugs/domains
urllist  drugs/urls
}

dest violence{
log  violence
domainlist  violence/domains
urllist  violence/urls
}

dest mail{
log  mail
domainlist  mai/domains
}

acl{
Test within Test_time {
 pass all
}
else{
pass !drugs !violence !mail  all
redirect http://www.yahoo.fr
}
default {
 pass none
}
}
```

- Création d'une nouvelle catégorie dans la liste noire :

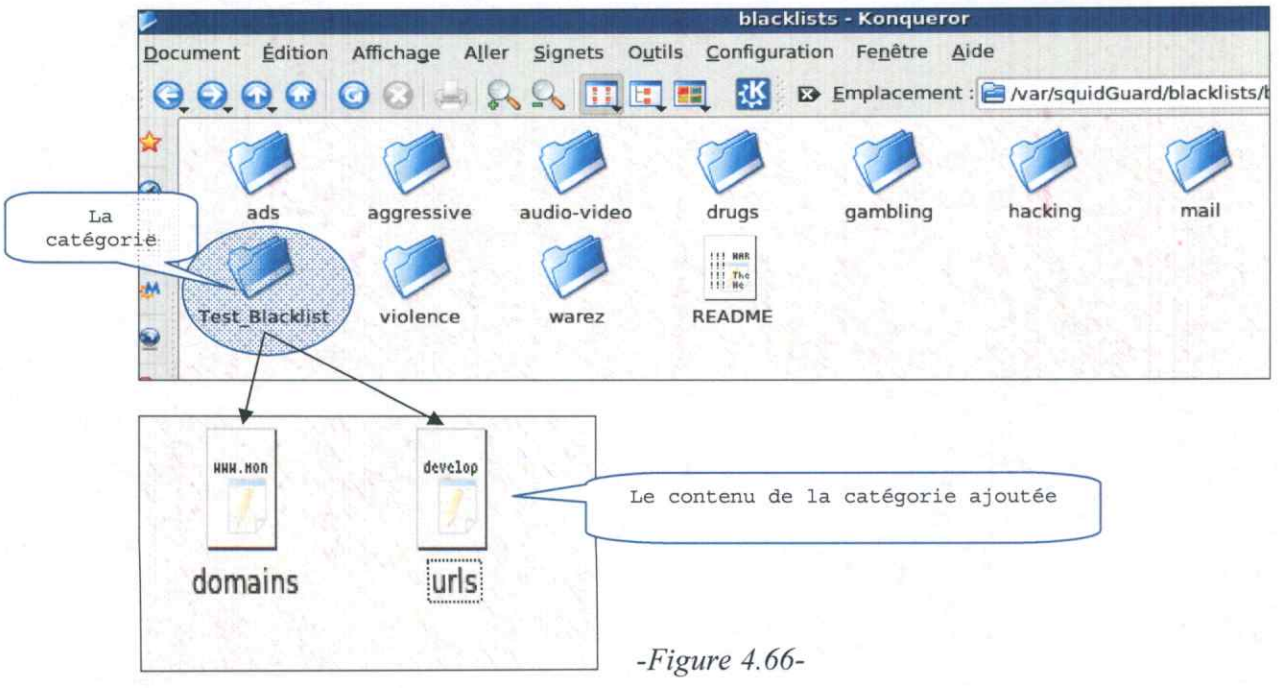
Pour ajouter une nouvelle catégorie dans la liste noire, il faut spécifier son nom et un domaine et une URL (ces deux champs sont obligatoires).



-Figure 4.65-

Test :

Une fois l'utilisateur valide, un nouveau dossier se crée avec deux fichiers domains, et urls qui vont contenir respectivement le domaine et l'url entrés précédemment.

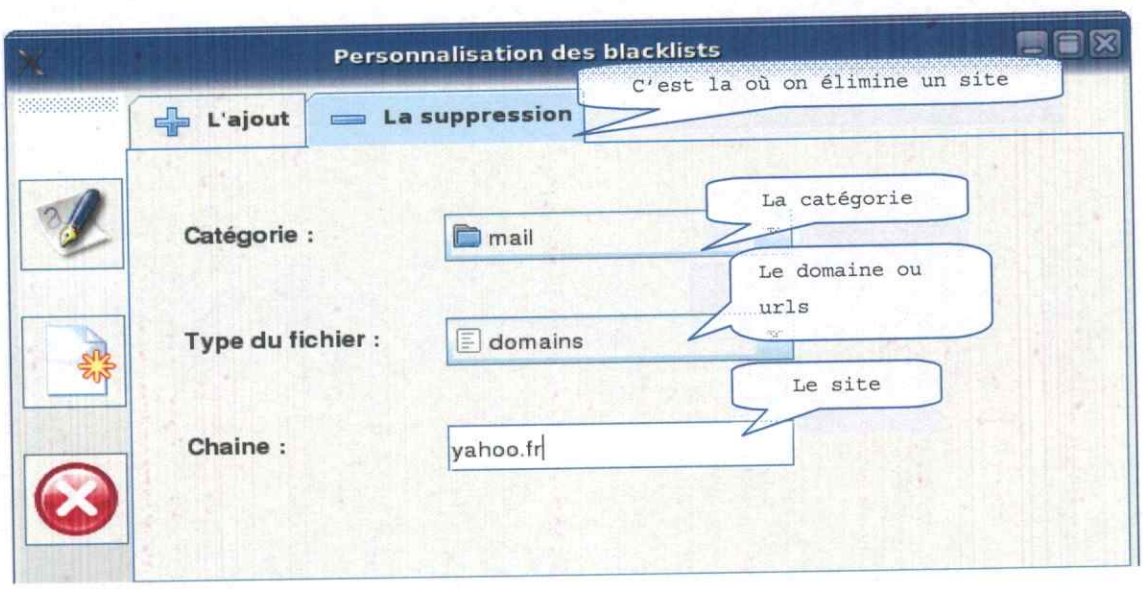


-Figure 4.66-

- Personnalisation de la liste noire :

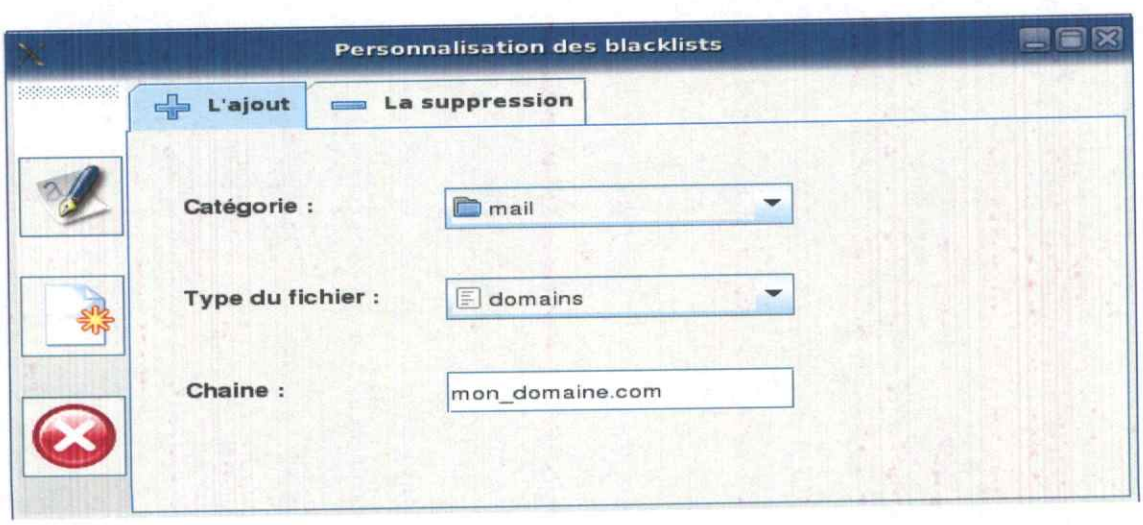
Parfois on veut autoriser un site qui se trouve dans une catégorie de la liste noire ou ajouter un site à la liste noire (que se soit un domaine ou une Url).

Prenons l'exemple de yahoo. Dans la catégorie mails et bien précisément dans le fichier domains il y a le domaine yahoo.com, pour l'éliminer de la liste noire c'est simple :



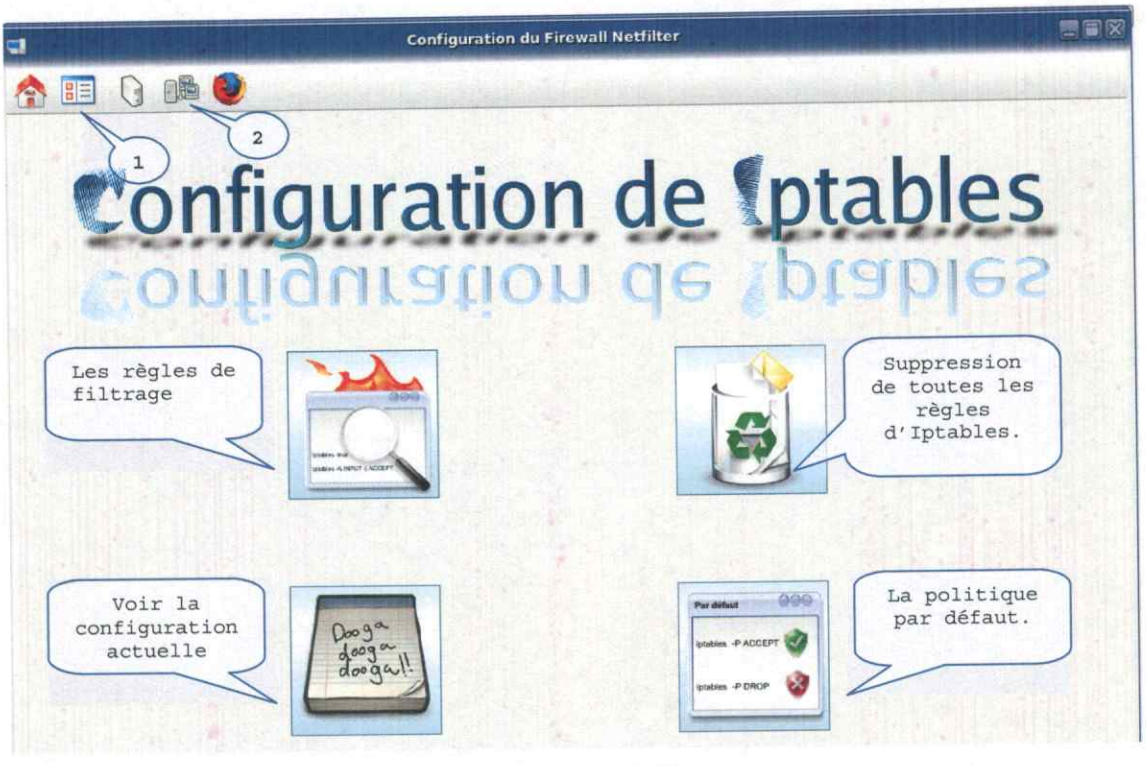
-Figure 4.67-

On procède de la même manière pour ajouter un site à liste noire.



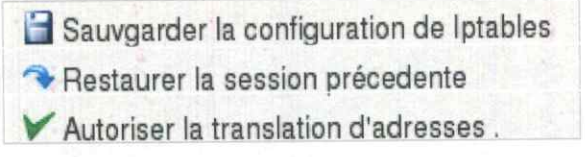
-Figure 4.68-

- Interface de configuration d'Iptables :



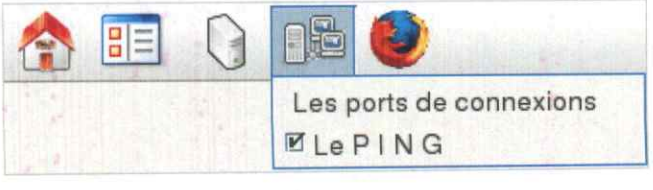
-Figure 4.69-

L'item N°=1 : permet de



-Figure 4.70-

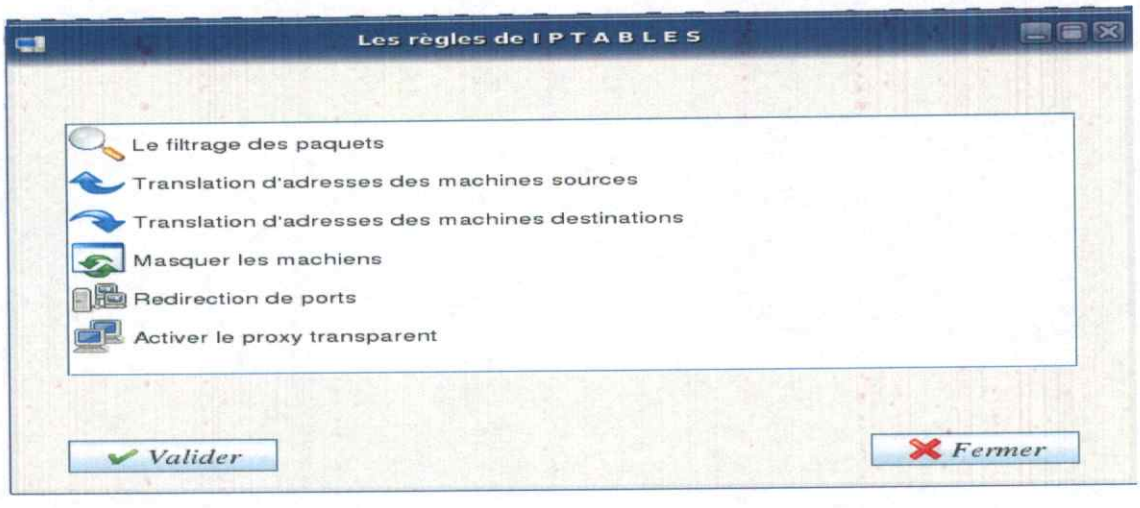
L'item N°=2 : permet de



-Figure 4.71-

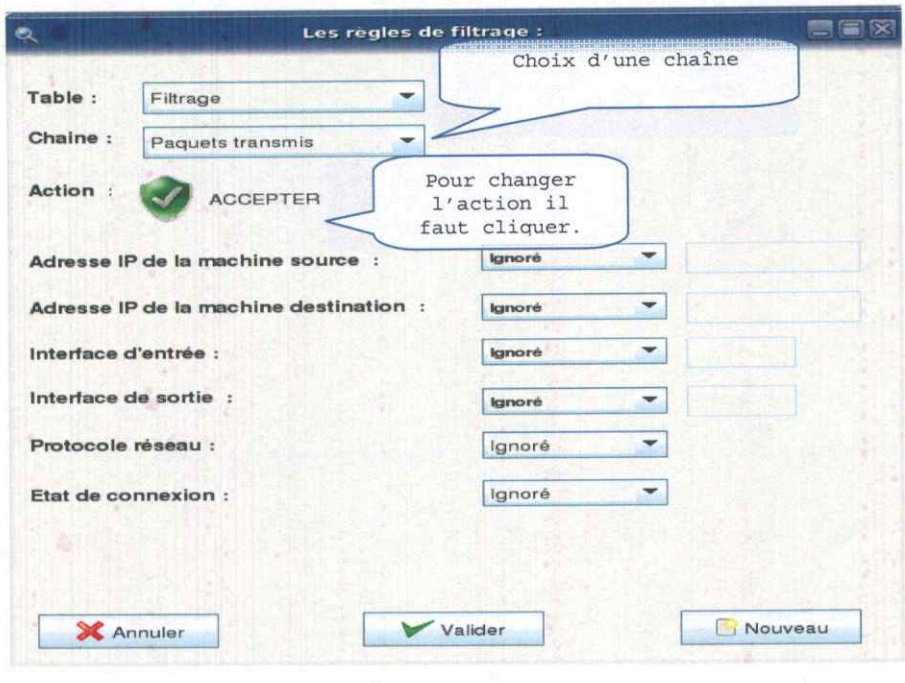
- Les règles de Filtrage :

Pour générer les règles d'Iptables l'utilisateur n'a qu'à sélectionner une règle parmi les suivantes :



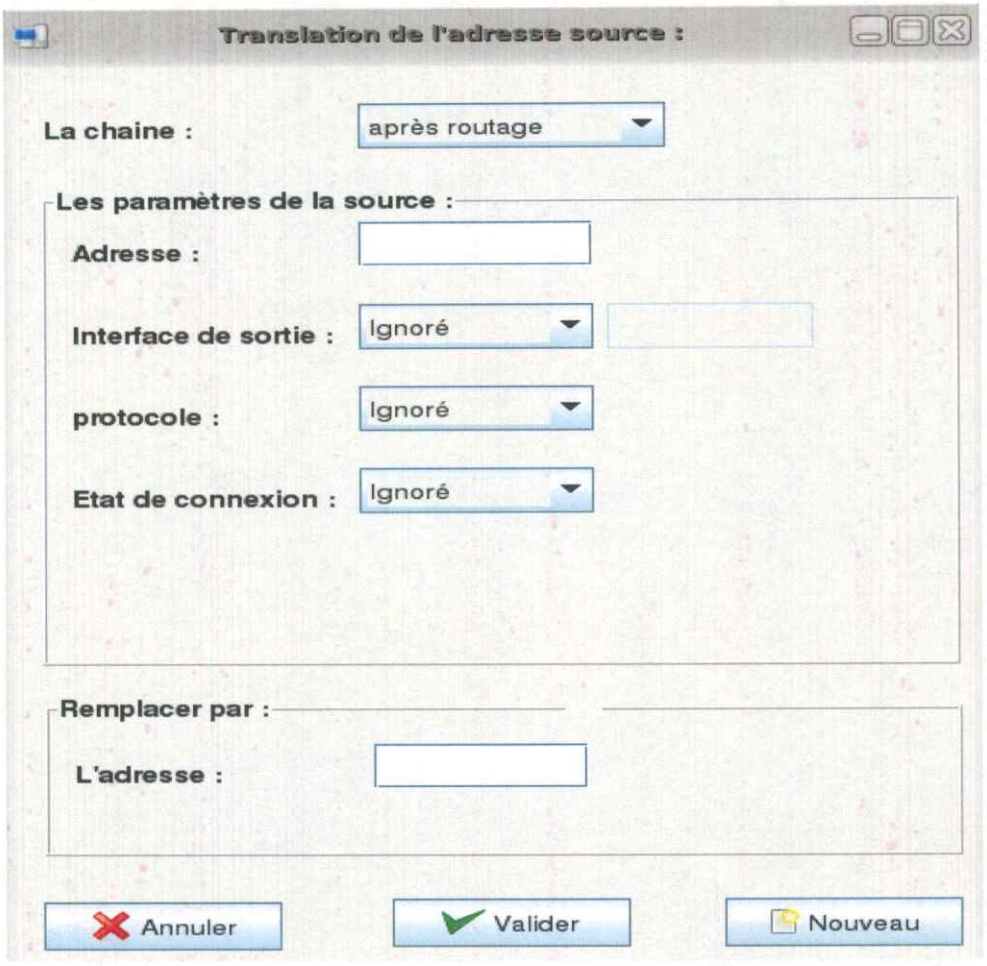
-Figure 4.72-

- Interface de Filtrage des paquets :



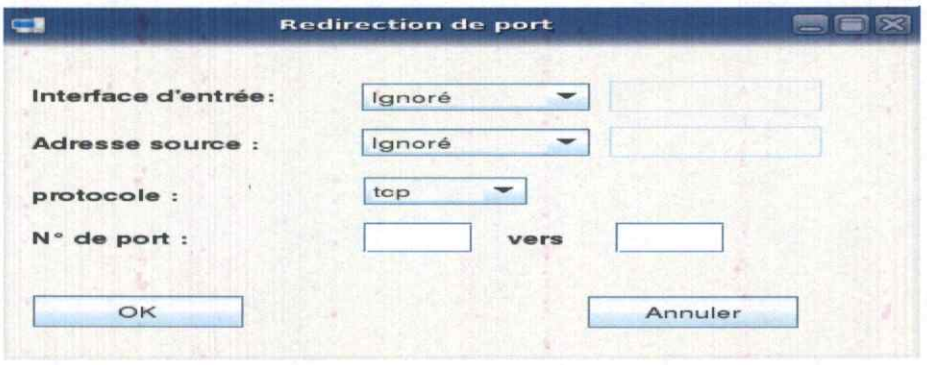
-Figure 4.73-

- Interface de translation d'adresses sources :



-Figure 4.74-

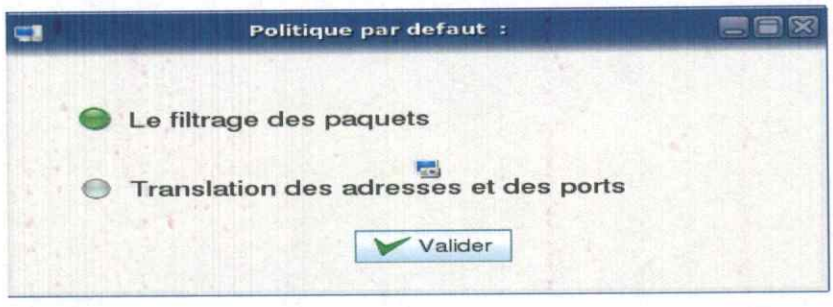
- Interface de redirection de ports :



-Figure 4.75-

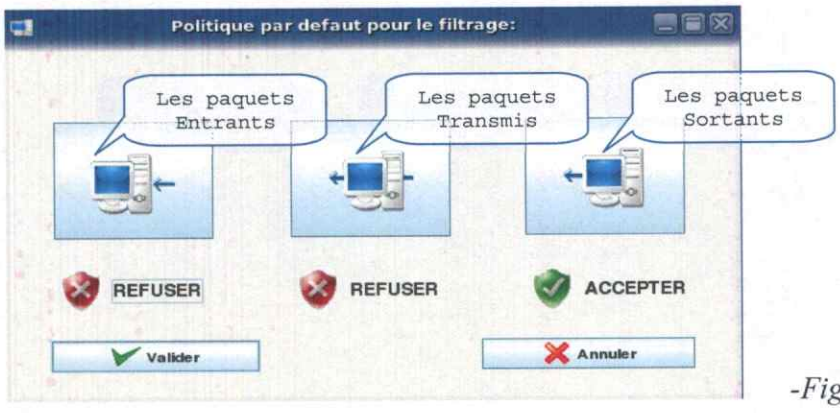
- Interface de personnalisation de la politique par défaut :

La politique par défaut et l'action effectuée lorsque aucune règle n'est applicable, notre outil permet de configurer les chaînes de la table FILTER ainsi que celles de la table NAT.



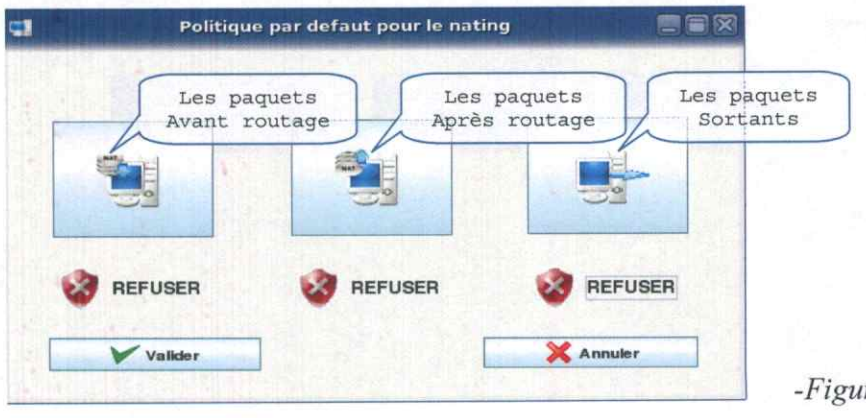
-Figure 4.76-

- Table FILTER :



-Figure 4.77-

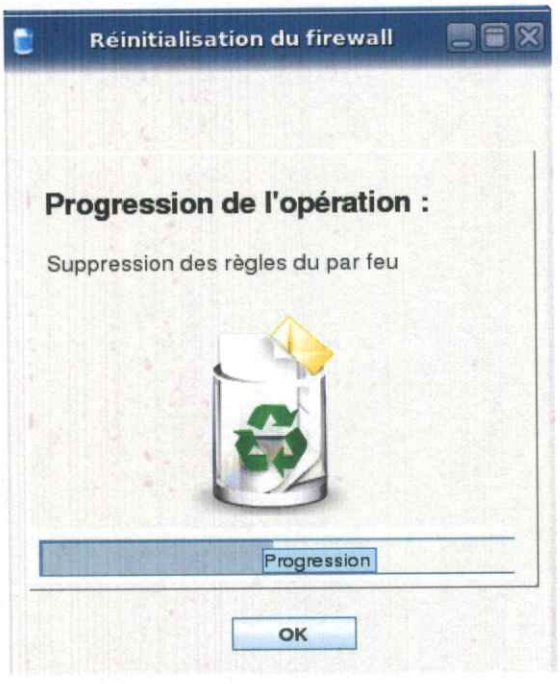
- Table NAT :



-Figure 4.78-

- Interface de suppression des règles d'IPTABLES :

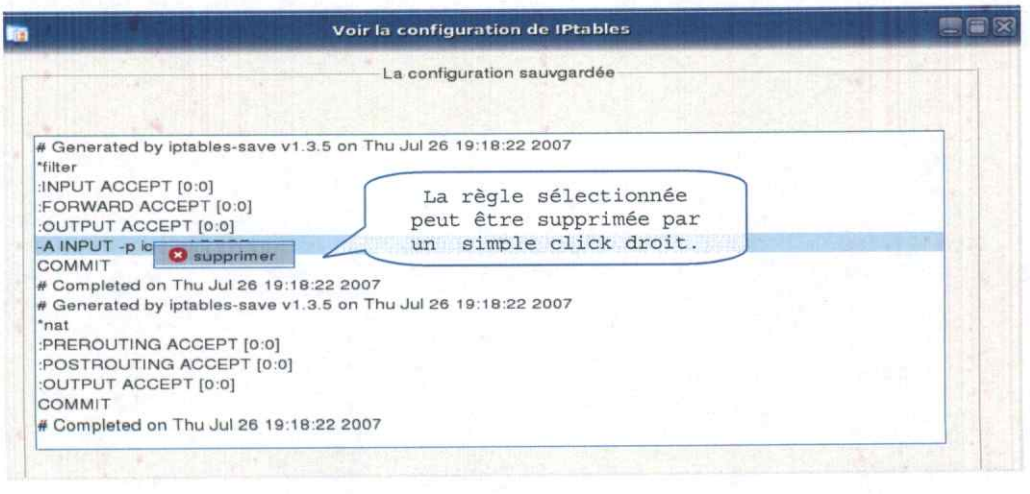
Pour effacer les règles d'IPTABLES et les rendre par défaut.



-Figure 4.79-

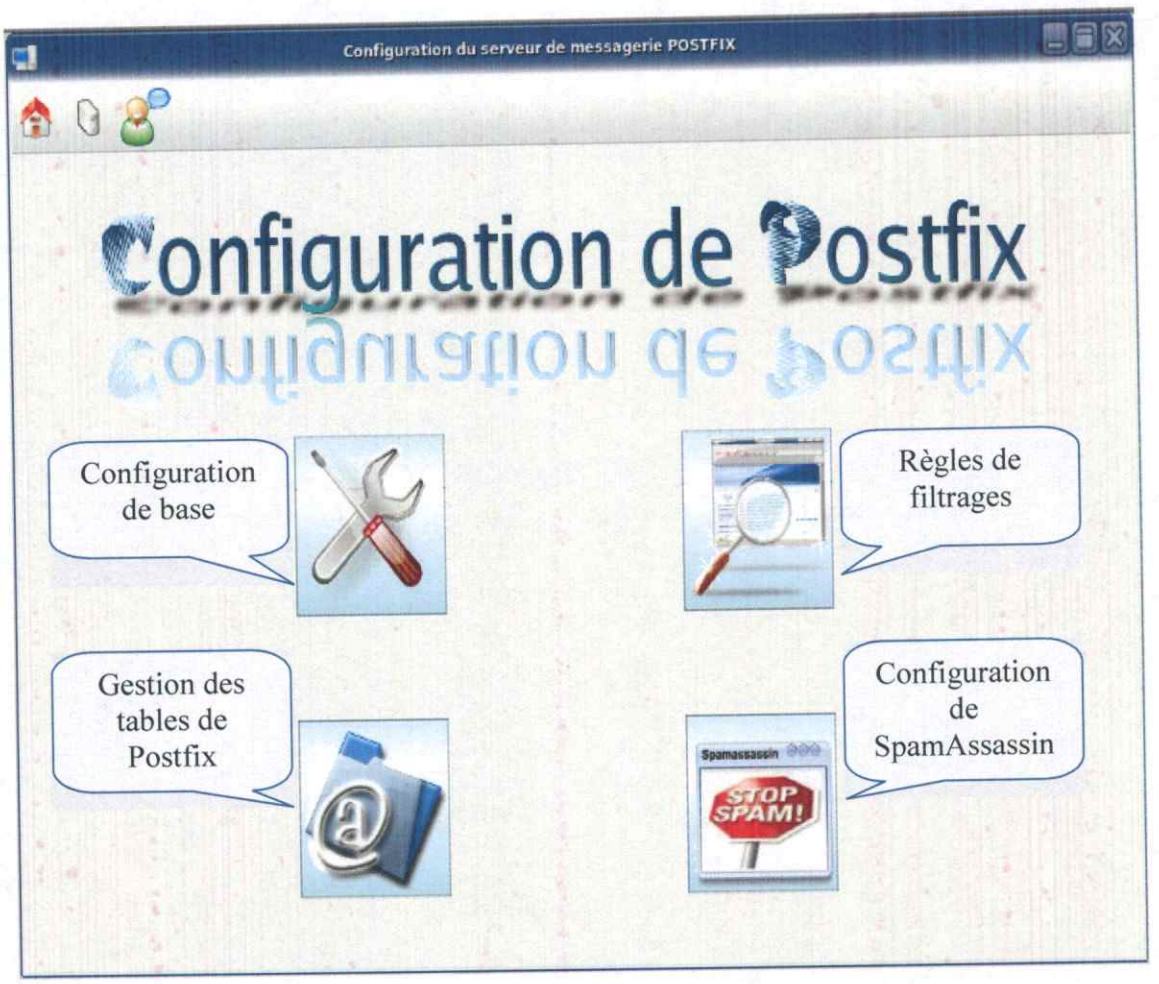
- Interface de visualisation des règles d'IPTABLES :

Cette interface visualise les règles sauvegardées d'IPTABLES.L'utilisateur pourra supprimer une règle par un simple click droit :



-Figure 4.80-

- Interface de configuration de Postfix :



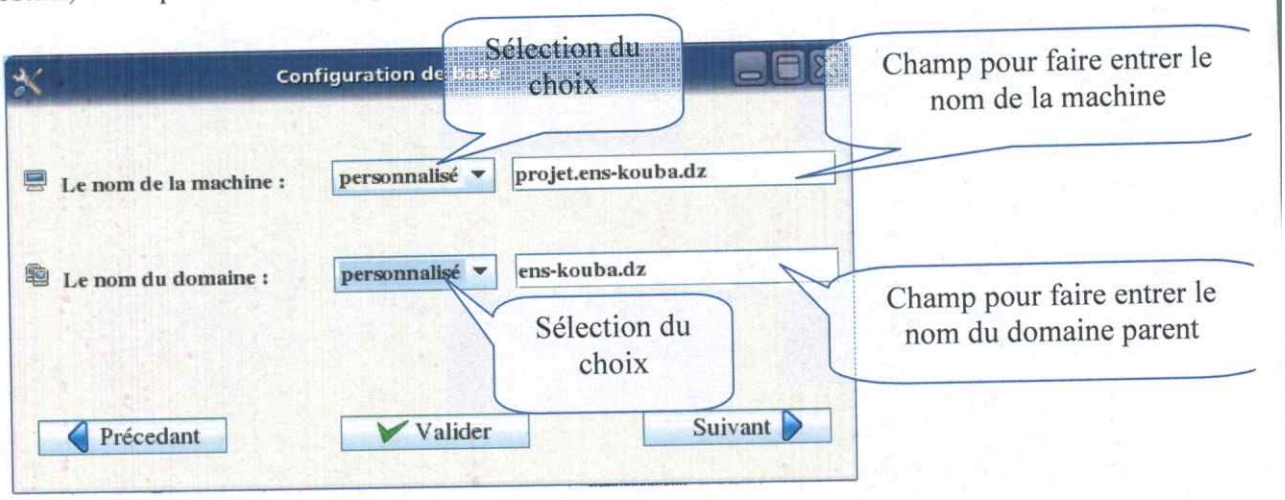
-Figure 4.81-

- Interface de configuration de base :

Notre outil affiche les valeurs des paramètres principales de Postfix, en donnant la possibilité à l'utilisateur de les modifier selon ses besoins.

- Interface de la première fenêtre de la configuration de base :

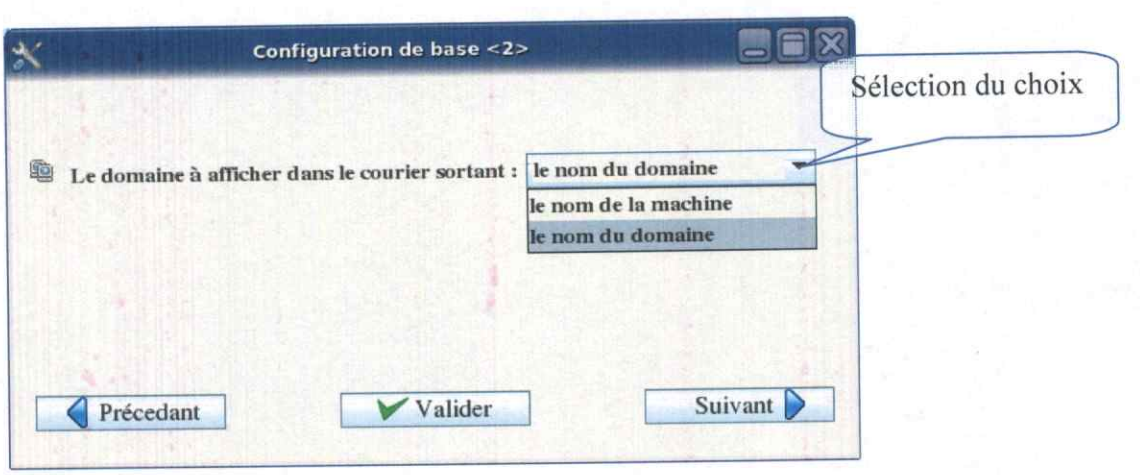
Dans cette fenêtre, L'utilisateur peut modifier le nom de la machine qui utilise Postfix, ainsi que son domaine parent.



-Figure 4.82-

- Interface de la deuxième fenêtre de la configuration de base :

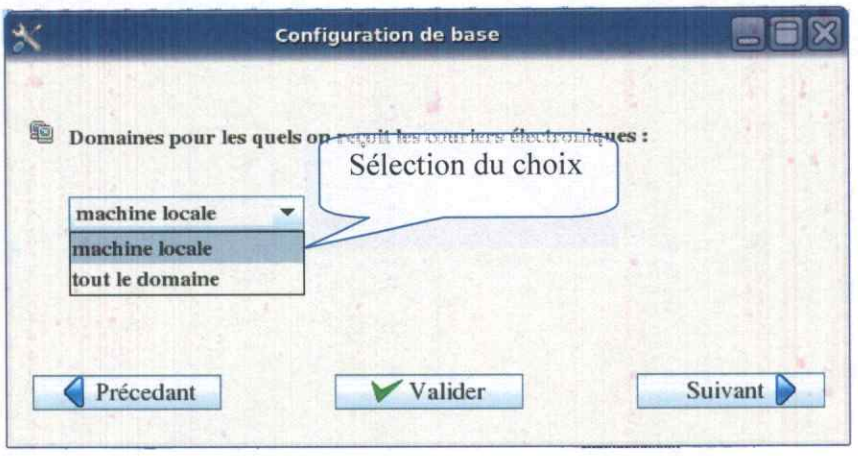
Dans cette fenêtre, l'utilisateur peut choisir le nom du serveur qui apparaîtra dans tout courrier posté sur cette machine, il a le choix entre le nom de la machine, et le nom de domaine de la machine.



-Figure 4.83-

- Interface de la 3 ème fenêtre de la configuration de base :

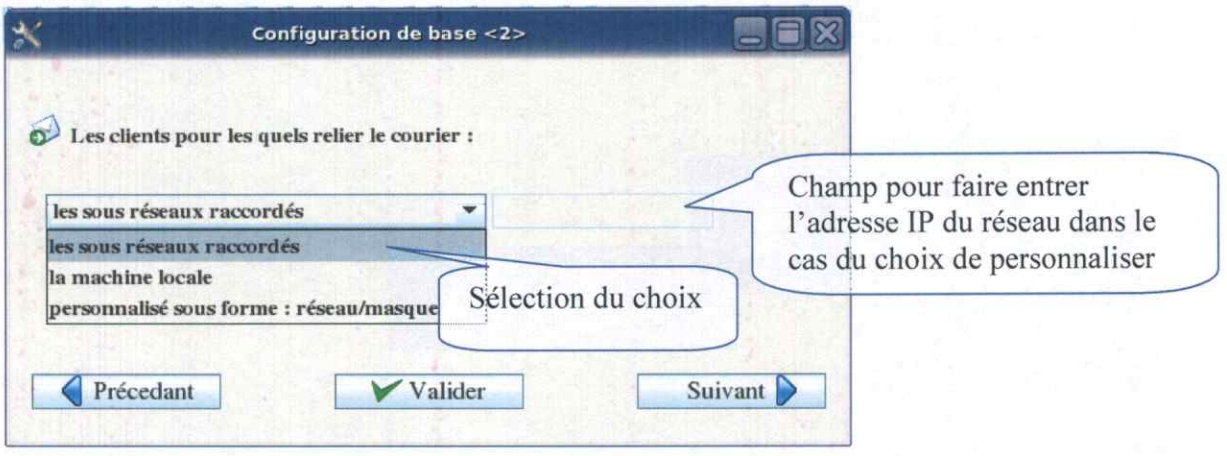
Dans cette fenêtre l'utilisateur peut choisir le domaine terminal pour l'acheminement des courriers électroniques.



-Figure 4.84-

- Interface de la 4 fenêtre de la configuration de base :

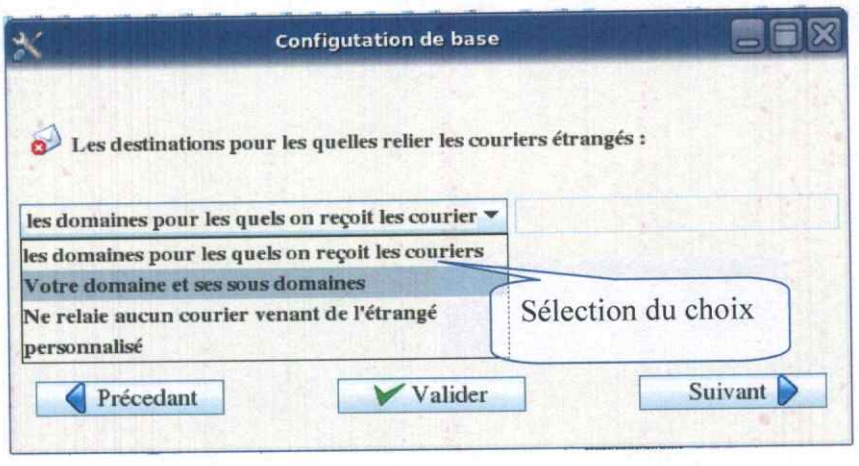
Dans cette fenêtre l'utilisateur peut choisir les clients qui sont autorisés à utiliser Postfix.



-Figure 4.85-

- Interface de la 5 ème fenêtre de la configuration de base :

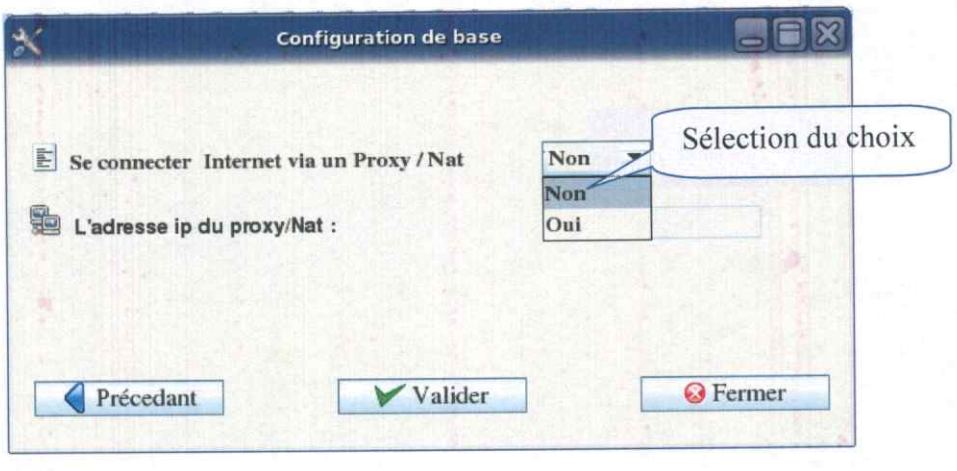
Dans cette fenêtre l'utilisateur définit les domaines pour les quels Postfix relaie le courrier étranger



-Figure 4.86-

- Interface de la 6 ème fenêtre de la configuration de base :

Dans cette fenêtre l'utilisateur peut choisir entre l'utilisation d'un Proxy/Nat, ou non.

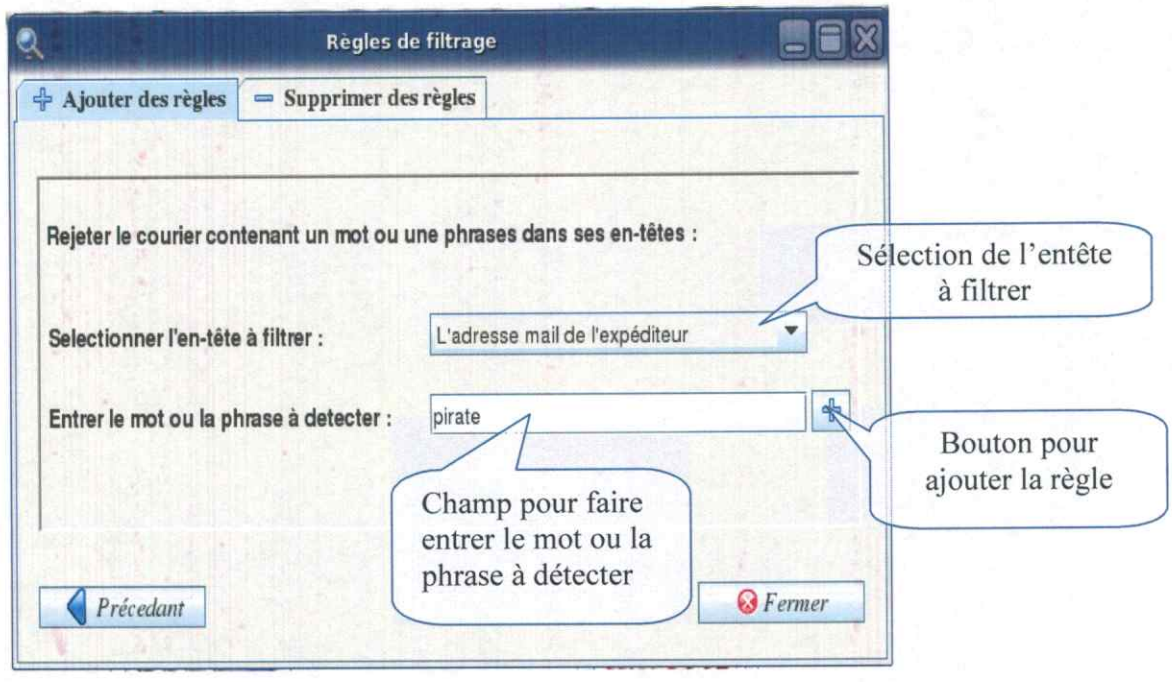


-Figure 4.87-

- Interface des règles de filtrages des entêtes des courriers électronique :

1) Interface pour l'ajout des règles de filtrage :

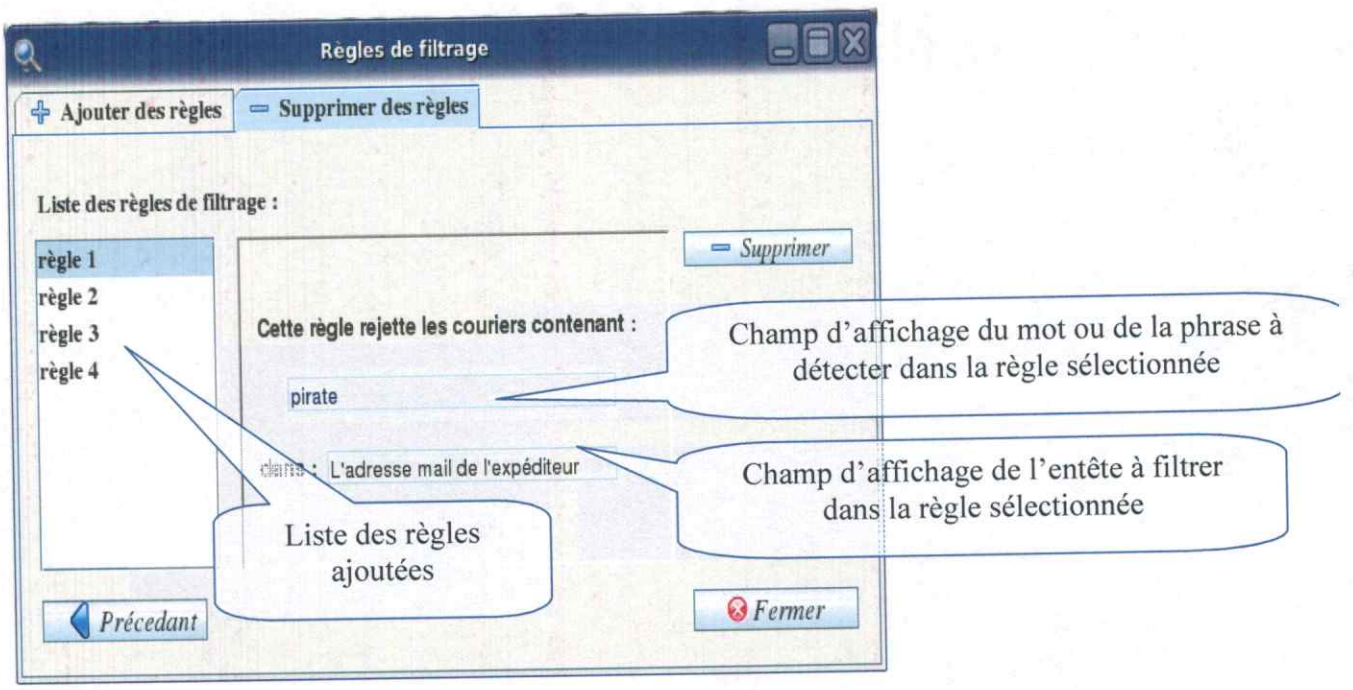
Notre outil permet l'ajout des règles de filtrage des entêtes, il suffira à l'utilisateur qu'à sélectionner l'entête qu'il veut filtrer, et faire entrer le mot ou la phrase à détecter, puis il valide en cliquant sur le bouton de l'ajout.



-Figure 4.88-

2) Interface pour la suppression des règles de filtrage :

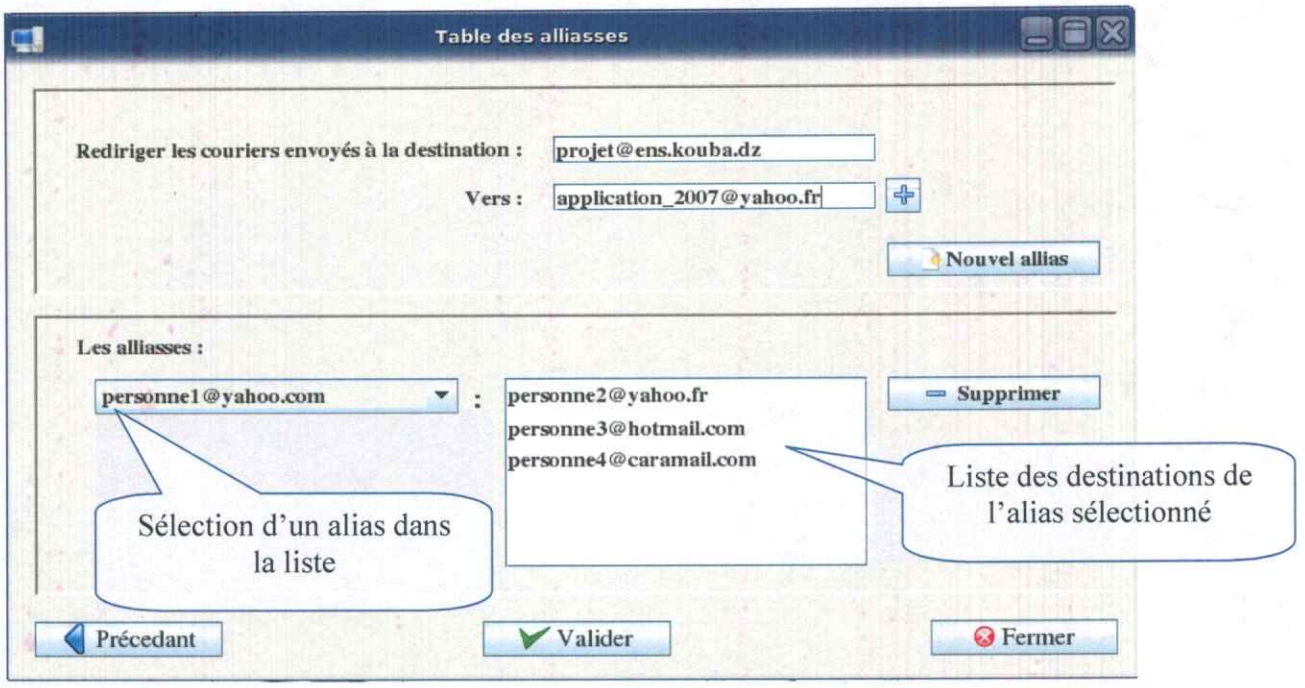
Notre outil permet la suppression des règles ajoutées, en affichant la liste des règles, lorsque l'utilisateur en sélectionne une, notre outil lui affiche ce qu'elle contient, si l'utilisateur voudra la supprimer, il n'a qu'à cliquer sur le bouton de suppression.



-Figure 4.89-

- Interface de la configuration de la table des alias :

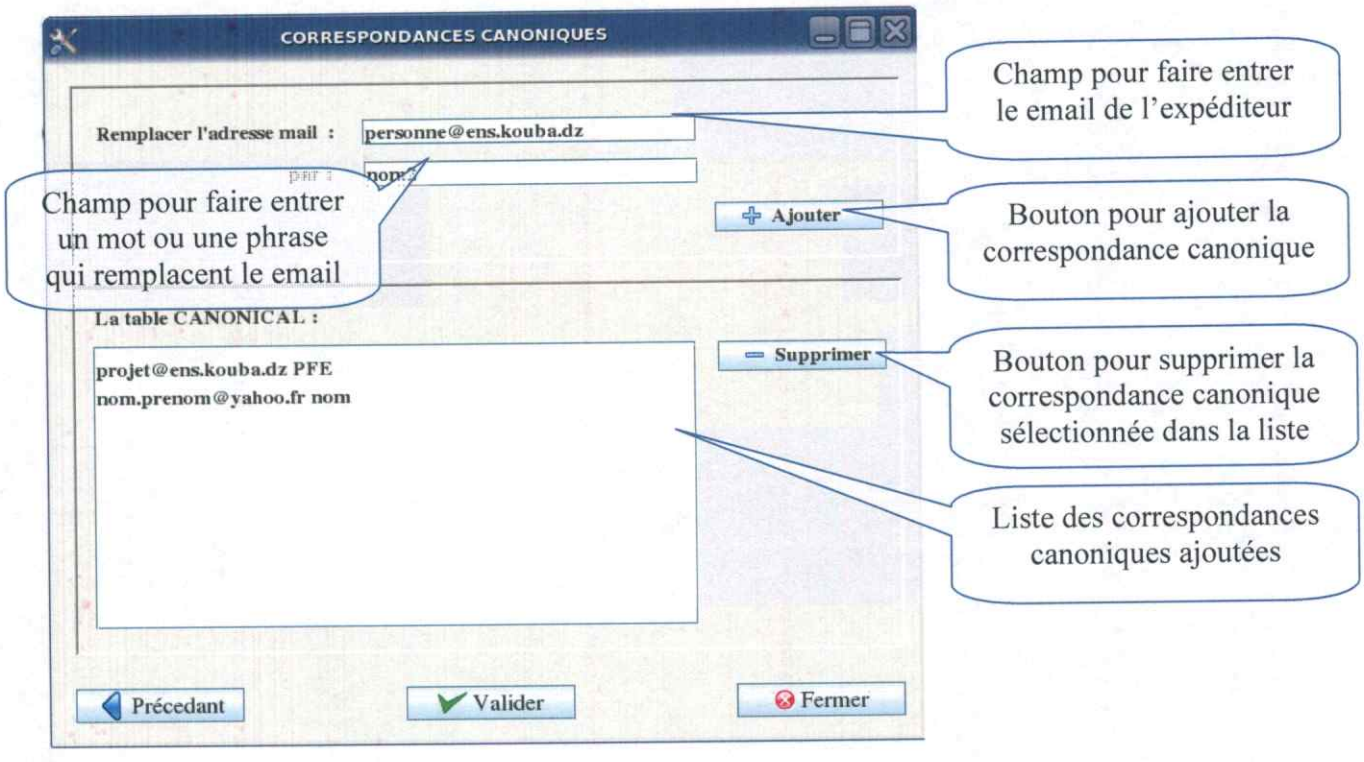
Notre outil permet de gérer la table des alias, pour cela l'utilisateur fait entrer l'adresse mail où il veut rediriger ses courriers vers d'autres destinations, et fait entrer ces dernières l'une après l'autre, il peut ajouter autant de destinations qu'il veut dans un alias.



-Figure 4.90-

- Interface de la configuration de la table canonical :

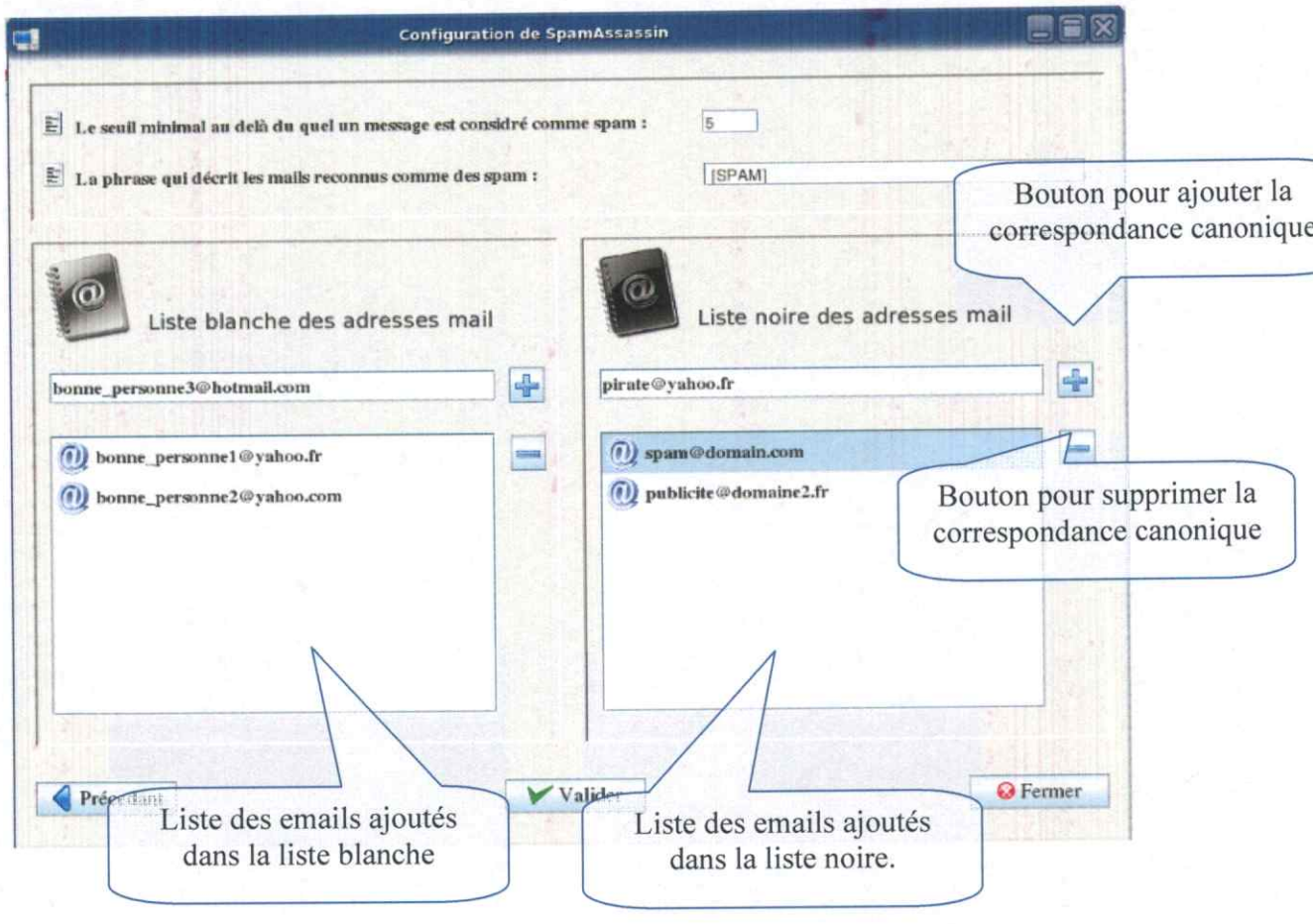
Notre outil permet de gérer la table canonical qui sert aux réécritures des adresses des expéditeurs, pour cela l'utilisateur fait entrer l'adresse mail de l'expéditeur, et un nom pour la remplacer.



-Figure 4.91-

Interface de la configuration de SpamAssassin :

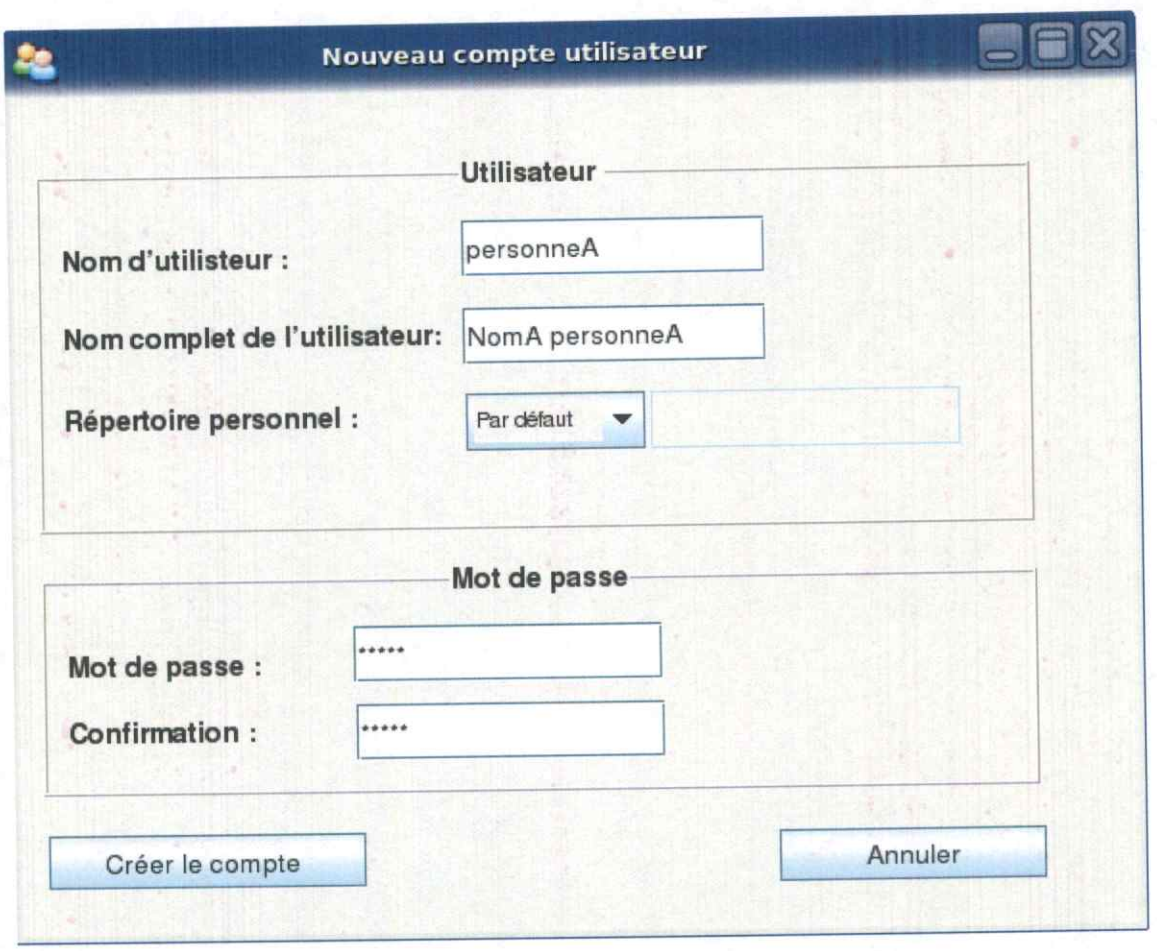
Notre outil permet de configurer SpamAssassin, l'utilisateur peut modifier les valeurs des paramètres de base (seuil minimal au-delà du quel les mails sont considérés comme du Spam, et la phrase à rajouter devant l'intitulé du message détecté comme du Spam), il peut aussi gérer les listes blanches des emails, et les listes noires.



-Figure 4.92-

- Interface de la création d'un compte utilisateur:

Il est possible de créer des comptes utilisateurs, pour cela il faut entrer le nom d'utilisateur, et le nom complet, puis faire entrer le mot de passe avec confirmation, ensuite valider.



-Figure 4.93-

CONCLUSION

L'objectif de ce travail a été de concevoir et de réaliser un outil d'aide à la configuration des différentes activités et services dans un réseau intranet pour simplifier la tâche d'administration.

Ce travail ne s'est pas déroulé sans difficultés. En effet, il était nécessaire de prendre connaissance de la configuration des divers services réseaux et de se familiariser avec les commandes d'administration et les fichiers systèmes pour assurer le bon fonctionnement des services qui ont fait l'objet de notre étude.

La réalisation a également été une tâche délicate, nous avons travaillé avec une alternative à la machine Java de Sun : **GCJ** (**G**NU **C**ompiler for **J**ava) couplé à une version dédiée d'Eclipse (Eclipse native) implémentés par défaut sous Fedora.

Malheureusement **GCJ** n'est pas encore assez mature et stable. Nous avons donc fait face à de nombreux bugs particulièrement avec les composants SWING et il nous a fallu écrire des classes à la main et chercher à chaque fois des alternatives pour pouvoir y remédier.

L'objectif principal de notre travail a été atteint. Nous avons pu réaliser un outil simple et ergonomique et offrant le nécessaire pour administrer un réseau.

Néanmoins, ce système est amené à évoluer et à prendre en compte la surveillance et la détection des anomalies de réseau, cela se fera par la réalisation d'un module permettant de :

- Reporter les divers états relatant les accès aux services.
- Les accès abusifs et leurs sources.

- Les tentatives d'accès internes ou externes au réseau.
- Les tentatives de passage de virus.
- Les accès à des services inexistants.
- Les accès non autorisés (autorisation totale ou suite à une authentification).

Nous pouvons dire que ce projet a été riche par les connaissances qu'il nous a permis d'acquérir. D'une part, de nous familiariser avec le système Linux et la diversité des services réseaux auxquels nous nous sommes intéressées et d'autre part, de maîtriser le langage Java et surtout son interface graphique.

BIBLIOGRAPHIE

[L_1] Grady Booch,James Rumbaugh & Ivar Jacobson
"*Le guide de l'utilisateur UML*" troisième édition, Ed Eyrolles 2003

[L _2] Cay S. Hortsmann & Gary Cornell
" *Au Coeur de Java 2 Notions fondamentales*", *Ed* CampusPress 2003

WEBOGRAPHIE

[WWW_1] http://www.apache.org/

[WWW_2] http://www.squid-cache.org/

[WWW_3] http://www.squidguard.org/

[WWW_4] http://www.netfilter.org/

[WWW_5] http://www.postfix.org/

[WWW_6] http://fedoraproject.org/

[WWW_7] http://mirrors.fedoraproject.org/publiclist/Fedora/6/

[WWW_8] http://gcc.gnu.org/java/

[WWW_9] http://home.golden.net/generator/

[WWW_10] http://www.wells.org.uk/htpasswdgen.html

[WWW_11] http://www.eclipse.org/

ANNEXE

A

Les protocoles réseaux

Les protocoles:

Dans les réseaux informatiques, un protocole de communication est une spécification de plusieurs règles pour un type de communication particulier.
Nous définissons dans le tableau ci-dessous les principaux protocoles cités dans ce document :

Protocole	définition
TCP	**Transmission Control Protocol** protocole de contrôle de transmissions, c'est un protocole de transport fiable, en mode connecté.
UDP	**User Datagram Protocol** Le rôle de ce protocole est de permettre la transmission de paquets de manière très simple entre deux entités. Il n'assure aucun contrôle de flux ni contrôle de congestion. C'est pour cela qu'il est souvent décrit comme étant un protocole non fiable. En revanche, pour un paquet UDP donné, l'exactitude du contenu des données est assurée grâce à une somme de contrôle (*checksum)*
ICMP	**Internet Control Message Protocol** Protocole de gestion des erreurs de transmission, de niveau 3 OSI. Il est utilisé par exemple quand vous faites un ping pour vérifier qu'une machine reliée au réseau est en état de fonctionner.
HTTP	**HyperText Transfer Protocol** protocole de transmission dédié aux clients et aux serveurs du web. HTTP utilise alors par défaut le port 80
FTP	**File Transfer Protocol** Protocole de transfert de fichier. Le FTP peut fonctionner en mode actif ou passif. Dans le premier cas, le client contacte le serveur sur le port 21, et celui-ci rappeler sur le port 20. Dans le monde passif, c'est le client qui initie toutes les connexions et seul le port 21 du serveur est utilisé.

Protocole	définition
SSL	**Secure Socket Layer** sockets sécurisées utilisées principalement par Netscape à l'origine, puis reconnues par l'ISOC, et s'étant généralisées dans leur version 3.
Gopher	mis au point par l'université du Minnesota pour la consultation d'informations organisées sous la forme d'une arborescence de menus hiérarchiques, fonctionnait en mode caractère. Il a disparu pour la simple et bonne raison que le protocole qu'il utilisait était la « propriété » de l'université du Minnesota
WAIS	**Wide Area Information Services** Ensemble de logiciels et de services conçus par *Brewster Kahle* permettant de rechercher des informations dans divers formats de fichiers à l'aide de mots-clés (ou de phrases en « langue naturelle »), et d'étendre la recherche dans les documents voisins. Ce service s'est éteint tranquillement, remplacé par les moteurs de recherche. En 1993, il était en position de quasi monopole dans le domaine sur l'Internet.
SMTP	**Simple Mail Transfer Protocol** protocole de la famille TCP/IP utilisé pour le transfert de courrier électronique. Utilisé évidemment sur l'Internet et reconnu par l'ISOC[1].
ESMTP	**Enhanced SMTP** Amélioration de SMTP, intégrant les types MIME et le DNS.
LMTP	**Local Mail Transfer Protocol** Protocole alternatif à ESTMP, permettant de délivrer du courrier localement sans gestion de file d'attente. Il ne doit pas être utilisé sur le port 25

Protocole	Définition
POP	**Post Office Protocol** Protocole de transfert de courrier électronique, prévu pour synchroniser les messages, et reconnu par l'ISOC[1]. Il en est à sa version 3 (incompatibles avec les précédentes), mais devrait être à terme remplacé par IMAP.
IMAP	**Internet Message Access Protocol** Protocole dans sa version 4, de gestion de messagerie, destiné à remplacer POP 3, qui est nettement moins performant. IMAP sait ainsi stocker le courrier sur le serveur et pas sur le client, et gérer toute la chose de façon correctement sécurisée.

1 : L'**Internet Society** (Isoc - http://www.isoc.org/) est une association de droit américain à vocation internationale créée en janvier 1992 par les pionniers de l'Internet pour promouvoir et coordonner le développement des réseaux informatiques dans le monde. Elle est en 2005 l'autorité morale et technique la plus influente dans l'univers du réseau Internet.