

L/005 - 02 / 01

Université Abou Bekr Belkaid

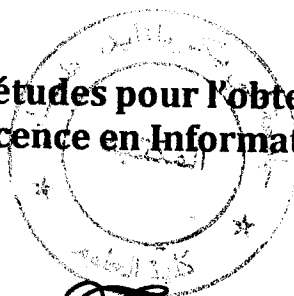


جامعة أبي بكر بلقايد

تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid- Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études pour l'obtention du diplôme
de Licence en Informatique



Thème

**Administration et sécurisation d'un
service mandataire squid et un
serveur web sur une machine
virtuelle**

Réalisé par :

- BENAIS Faiza
- YELLES CHAUCHE Sarra Nassira

Présenté le 27 Juin 2013 devant la commission d'examination composée de MM.

- *M^{me} DIDI Fedoua* (Examineur)
- *M^{me} LABRAOUI Nabila* (Examineur)
- *Mr BENAISSA Mohammed* (Encadreur)



Année universitaire : 2012-2013

SOMMAIRE

Introduction générale	1
-----------------------------	---

Chapitre N°1 : Introduction à la virtualisation

1. Introduction	2
2. Évolution de la virtualisation	2
3. La machine virtuelle	4
4. Intérêt de la virtualisation	6
5. Contrainte de la virtualisation	6
6. Technologies de la virtualisation	7
6.1. La virtualisation complète ou machine virtuelle	7
6.2. La para virtualisation	8
6.3. Le Système a Hyperviseur	9
7. Logiciel de virtualisation : Virtualbox	11
7.1. Configuration réseau de virtualbox	12
8. Conclusion	13

Chapitre N°2 : Administration d'un serveur web et un serveur ftp

1. Introduction	14
2. Serveur WEB	14
3. Serveur web apache	16
3.1. Le protocole http utilisé par web	16
3.2. Installation et configuration apache	17
4. Serveur ftp	19
4.1. Protocole FTP	20
4.2. Mode de fonctionnement d'un serveur ftp	20
4.3. Configuration d'un serveur vsftp sous linux	21
5. Conclusion	23

Chapitre N°3 : Configuration d'un serveur proxy squid

1. Introduction	24
2. Les avantages d'un proxy	24
2.1. Utilisation d'un cache	24
2.2. Filtrage	25

2.3. Authentification client	25
2.4. Reverse proxy	26
3. Limitations	26
4. Protocoles supportés	26
5. Installation et configuration de SQUID	27
5.1. Administration et configuration d'un squid	27
5.2. Les Contrôles d'accès (ACL)	29
5.3. Installation et configuration de SQUIDGUARD	31
5.4. Installation des listes noires	31
5.5. Paramétrer SquidGuard	31
5.6. Configuration squidguard	32
5.7. Compiler les blacklists	33
6. Conclusion	33

Chapitre N°4 : Administration proxy squid par webmin

1. Introduction	34
2. Architecture de notre réseau virtuel	34
3. Administration du serveur proxy squid par webmin	36
3.1. Présentation de Webmin	36
3.2. Configuration des ports et de l'interface d'écoute	38
3.3. Configuration du Cache	38
4. Définition des ALC	39
4.1. Les contrôles d'accès au proxy squid	39
5. Définition du réseau local	40
6. Définition des ALCs pour les heures de connexions	40
7. Application des ACLs	41
7.1. Définition de l'authentification du nom d'utilisateur / mot de passe...	42
8. Configuration de squidguard	45
9. Conclusion	48
Conclusion générale	49
Bibliographies	50
Liste des figures	51
Liste des abréviations	52

Squid est un proxy assez performant. De base qu'il permet déjà beaucoup de choses : fermer des ports un par un, empêcher des plages d'adresse ip de se connecter à internet via ce proxy.

Mais ce n'est qu'avec le plugin Squidguard que celui-ci permet de filtrer une à une les URL en fonction des blacklists paramétrables.

Donc en premier lieu on va s'intéresser à la virtualisation et cela en utilisant l'outil virtual box qui est un logiciel très performant permettant de faire tourner un système dans un autre, tout cela de manière « virtuelle ». Ensuite on va procéder à l'installation d'un serveur Web qui est un ordinateur connecté en permanence à l'Internet, apache qui est un serveur web http libre, car c'est un des serveurs web le plus utilisé sur Internet avec plus de 60% des sites d'Internet.

Enfin le File Transfer Protocol (protocole de transfert de fichiers), ou FTP, qui est un protocole de communication destiné à l'échange de fichiers sur un réseau TCP/IP.

Finalement pour aider notre proxy à devenir plus simple que via le config on l'a couplé avec une application qui s'appelle Web min et qui est un gestionnaire graphique de serveurs suivie de règles de restrictions.

Chapitre N°1

Introduction à la virtualisation

1. Introduction

Il y a quelques années, l'imagination de l'industrie informatique s'est emparée de l'idée consistant à faire fonctionner plusieurs systèmes d'exploitation de manière concurrente sur un même ordinateur. La « virtualisation »¹ est devenue le mot à la mode et des projets ont été lancés pour réaliser ce rêve.

Grace à la virtualisation, vous n'avez pas besoin d'un ordinateur supplémentaire chaque fois que vous voulez mettre en place un nouveau serveur. On peut faire face à des besoins supplémentaires en termes d'infrastructure en démarrant simplement un nouveau système d'exploitation invité. Un système invité peut être dédié à une application unique et il peut être différent du système d'exploitation hôte. Les fonctionnalités telles que la virtualisation de stockage permettent de déplacer les systèmes invités sans interruption de l'activité afin d'exploiter au mieux votre matériel informatique.

La virtualisation est devenue une solution d'entreprise qui permet de réduire le nombre des serveurs physiques. Mais, par contre, elle permet d'augmenter conséquemment le nombre des serveurs virtuels sur chaque serveur physique, en vue d'optimiser son utilisation, de réduire les dépenses sur le matériel serveur, de diminuer la consommation électrique ainsi que de libérer beaucoup d'espace dans la salle serveur en facilitant l'administration du système informatique.

Elle peut être déployée dans un réseau d'entreprise, intranet, qui se définit comme tout réseau TCP/IP privé, qui emploie les mêmes technologies, services que ceux de l'Internet.

2. Évolution de la virtualisation

Les premiers ordinateurs, qui occupaient plusieurs pièces d'un bâtiment, n'étaient pas faits pour exécuter *plusieurs* programmes à la fois. On concevait un programme (qui était à l'époque une simple succession de calculs), on le mettait dans une file d'attente des programmes, et quand le système d'exploitation avait fini de traiter un programme, on lui donnait le suivant dans la liste.

Très vite, dès la fin des années cinquante, l'idée de pouvoir exécuter plusieurs programmes en parallèle voit le jour. On parle de temps partagé (*time sharing*), de multiprogrammation, etc. L'idée était de pouvoir faire cohabiter plusieurs programmes

¹ Eric Maillé, Damien Bruley et René-François, Les solutions de virtualisation au sein de votre organisation (serveur et poste de travail), Eni, 2012

au même moment, ayant tous accès au même matériel, sans qu'ils ne se gênent mutuellement.

La virtualisation est très proche du concept.

Les systèmes invités étaient gérés par une simple multiprogrammation. En 1967 est lancé, toujours par IBM, le système CP-40, le premier système offrant une virtualisation complète. Le CP-40 sera suivi par plusieurs évolutions, amenant chacune de nouvelles fonctionnalités pour les utilisateurs. On peut notamment citer le système VM/370, qui a connu un très fort succès dans les entreprises, et est parfois encore en usage dans certaines entreprises aujourd'hui.

Après le succès des machines virtuelles introduites par IBM, les technologies ont assez peu évolué. Le système hôte a vite été réduit à l'état de simple arbitre entre les systèmes invités, amenant la notion d'hyperviseur. Toutefois, toutes ces technologies de virtualisation étaient réservées au monde professionnel, destinées à être utilisées sur des *mainframes* coûtant plusieurs millions de dollars.

Parallèlement à cela, le monde de la recherche (souvent financé par ces mêmes entreprises) a continué à étudier différentes possibilités pour améliorer les performances et à essayer de nouvelles technologies. La plupart de ces travaux de recherche sont toutefois restés assez confidentiels et n'ont que rarement été transposés sur un produit.

L'orientation « grand public » des technologies de virtualisation est beaucoup plus récente. Dans les années quatre-vingt-dix, l'intérêt pour les émulateurs de consoles de jeu ainsi que l'explosion du marché de l'informatique personnelle (les ordinateurs de type PC) ont fait prendre conscience aux entreprises qu'il y avait un marché pour la virtualisation sur PC. Des sociétés ont alors commencé à créer des produits de virtualisation basés sur des machines virtuelles pour les « petites » entreprises c'est à dire celles ne pouvant s'offrir des serveurs à plusieurs millions de dollars et pour les particuliers.

Prenons l'exemple d'une solution de virtualisation faite pour le grand public, de type VMware ou virtualbox : l'utilisateur possède un seul ordinateur, sur lequel est installé un système d'exploitation (Microsoft Windows, GNU/Linux, Mac OS X, etc.) ainsi qu'une application qui fait office de machine virtuelle : le logiciel installé par VMware. L'utilisateur peut à partir de son système d'exploitation appelée aussi système hôte, démarrer un nouveau système d'exploitation qui peut être totalement différent de celui installé sur la machine physique.

Le système d'exploitation virtualisé appelé système invité (guest system) est alors exécuté par la machine virtuelle, et complètement détaché de tout le matériel de l'ordinateur. La machine virtuelle se charge d'émuler pour le système invité tout le matériel « standard » d'un ordinateur : disque dur, écran, clavier, souris, etc. L'utilisateur peut alors utiliser le système invité comme normal : installer l'application, naviguer sur Internet, exécuter un programme, etc. Le système hôte installé sur la machine physique et le système invité sont totalement indépendants : le système invité est vu par l'hôte comme un simple programme, il n'a pas d'accès direct au matériel contrairement à l'hôte.

Aujourd'hui, les solutions de virtualisation couvrent principalement deux domaines : les systèmes de stockage et les systèmes serveurs. Dans le cas des environnements de stockage, la virtualisation est utilisée par les administrateurs pour gérer les différentes ressources par une vue unique virtuelle. Par cette vue, ils gèrent de façon centralisée l'espace disponible indépendamment des technologies utilisées. La virtualisation des systèmes serveurs répond à un objectif similaire. Les entreprises en ont recours pour disposer d'une vue générale sur l'utilisation de leurs ressources machines. Ils peuvent ainsi découper un serveur physique en des multiples serveurs logiques, dont chacun se verra attribuer une tâche différente. Les vues virtuelles ne tiennent pas compte des technologies exécutées par la machine.

3. La machine virtuelle

Une machine virtuelle est un ordinateur logiciel qui, à l'instar d'un ordinateur physique, exécute un système d'exploitation et des applications. La machine virtuelle se compose d'un ensemble de fichiers de spécification et de configuration ; elle est secondée par les ressources physiques d'une hôte. Chaque machine virtuelle a des périphériques virtuels qui fournissent la même fonction que le matériel physique et présentent un intérêt supplémentaire en terme de portabilité, maniabilité et sécurité.

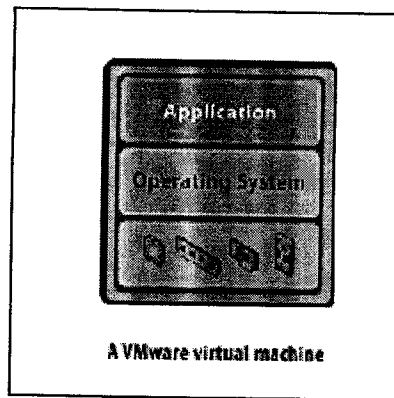


Figure 1.1 : machine virtuelle

Un ordinateur virtuel est constitué par des composants suivants :

- Une fenêtre dans laquelle l'ordinateur virtuel s'exécute. Chaque ordinateur virtuel est doté de paramètres qui déterminent sa relation de fonctionnement avec le système d'exploitation hôte, tels que la quantité de mémoire vive allouée à l'ordinateur virtuel ou les affectations des ports COM1 et COM2. Virtual PC possède également des options globales qui affectent tous les ordinateurs virtuels.
- Un fichier de configuration d'ordinateur virtuel « (.v`mc`) »² qui contient toutes les informations de configuration d'un ordinateur virtuel.
- Un fichier qui représente le disque dur de l'ordinateur virtuel, appelé disque dur virtuel. Les disques durs virtuels ont une extension « .v`hd` »³ Par défaut, ce disque est un fichier de taille dynamique dont la taille augmente à mesure que vous y installez des applications ou que vous y stockez des données. Vous pouvez sélectionner différents types de disques durs virtuels et configurer pour l'ordinateur virtuel jusqu'à trois disques durs virtuels différents, afin qu'il bénéficie de plus d'espace de stockage.
- Un système d'exploitation installé sur le disque dur virtuel. Il peut s'agir quasiment de n'importe quel système d'exploitation compatible avec les ordinateurs x86.
- Du matériel émulé et des périphériques externes, tels que clavier, souris, CD, DVD, disquette, carte audio, ports, imprimantes et autres périphériques utilisés par l'ordinateur virtuel pour émuler un ordinateur physique.

² Tunisien team ; Wajih Letaief Virtualisation sous Ubuntu avec VirtualBox novembre 2008
³ ibid

4. Intérêt de la virtualisation

Les intérêts de la virtualisation sont :

Utilisation optimale des ressources d'un parc de machines c'est-à-dire la répartition des machines virtuelles sur des machines physiques en fonction des charges respectives ;

Installation, déploiement et migration facile des machines virtuelles d'une machine physique à une autre, notamment dans le contexte d'une mise en production à partir d'un environnement de qualification ou de pré production, livraison facilitée ;

Économie sur le matériel par mutualisation : consommation électrique, entretien physique, monitoring, support, compatibilité matérielle, etc. ;

Installation, tests, développements, réutilisation avec possibilité de recommencer, arrêt du système hôte sans déranger les autres machines ;

Sécurisation et /ou isolation d'un réseau (arrêt des systèmes d'exploitation virtuels, mais pas des systèmes d'exploitation hôtes qui sont invisibles pour l'attaquant, tests d'architectures applicatives et réseau) ;

Isolation des différents utilisateurs simultanés d'une même machine ;

Allocation dynamique de la puissance de calcul en fonction des besoins de chaque application à un instant donné

Diminution des risques liés au dimensionnement des serveurs.

5. Contrainte de la virtualisation

Bien que la virtualisation nous amène plusieurs avantages, nous allons évoquer ici quelques inconvénients :

- Plusieurs serveurs virtuels sont sur une seule machine physique, C'est un risque potentiel ;
- Si une VM a un problème de performance (100% CPU) alors les autres serveurs sont affectés (plus ou moins)
- Perte de performance pour les disques durs
- La facilité de création de VM implique une surconsommation des ressources

Ces inconvénients peuvent bien sûr être réglés par l'application de bonnes règles de gestion des machines virtuelles.

6. Technologies de la virtualisation

Il existe plusieurs catégories de virtualisation, utilisant chacune des technologies différentes. Les technologies les plus répandues sont :

- la virtualisation complète ou machine virtuelle ;
- la para virtualisation ;
- la virtualisation assistée par le matériel ou hyperviseur ;

Chacune de ces technologies est une technologie de virtualisation, mais elles ne fonctionnent pas de la même façon. Les principes et particularités de chaque technologie seront détaillés comme suit.

6.1. La virtualisation complète ou machine virtuelle

La virtualisation complète en anglais « full virtualisation » est une technologie qui consiste à émuler l'intégralité d'une machine physique pour le système invité. Le système invité « croit » s'exécuter sur une véritable machine physique. Le logiciel chargé d'émuler cette machine s'appelle « une machine virtuelle », son rôle est de transformer les instructions du système invité en instruction pour le système hôte. Ainsi, la machine virtuelle est un programme comme un autre du point de vue du système hôte, au même titre qu'un navigateur Internet ou traitement de texte.

Les programmes utilisateur n'ont pas d'accès direct au matériel, mais uniquement aux couches d'abstraction. La machine virtuelle émule donc de manière logique tout le matériel habituel de l'architecture pour le système invité, ce dernier croit dialoguer directement avec l'édit matériel.

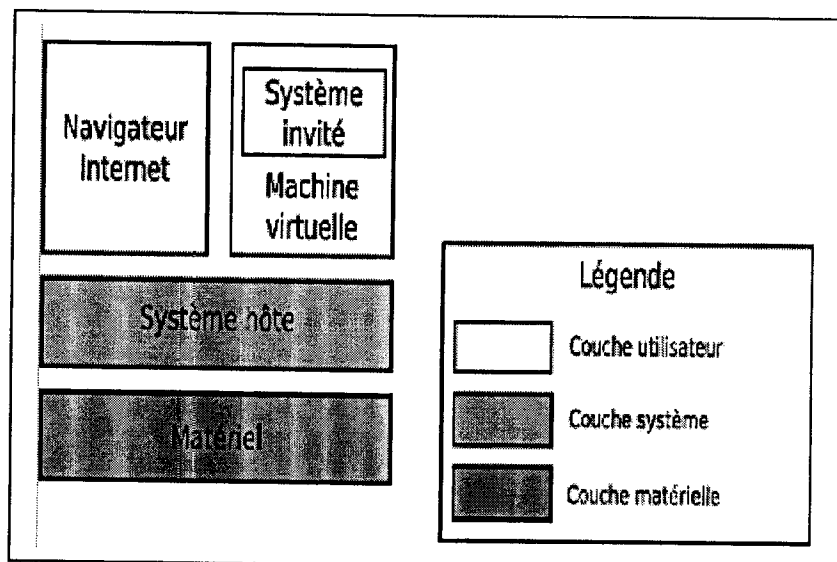


Figure 1.2 : virtualisation complète

Cet empilage de couches est sensiblement identique pour tous les périphériques émules par la machine virtuelle. On retrouve. Du plus bas au plus haut niveau :

1. Le matériel.
2. Le pilote du matériel pour le système hôte,
3. La couche d'abstraction du système hôte,
4. Le matériel émulé par la machine virtuelle,
5. Le pilote du matériel pour le système invité,
6. La couche d'abstraction du système invité,

6.2. La para virtualisation

La « paravirtualisation »⁴ (*paravirtualization* ou en core *paravirtualization*) est très proche du concept de la virtualisation complète, dans le sens où c'est toujours un système d'exploitation complet qui s'exécute sur le matériel émulé par une machine virtuelle, cette dernière s'exécutant au-dessus d'un système hôte. Toutefois, dans une solution de paravirtualisation,

le système invité est modifié pour être exécuté par la machine virtuelle. Les modifications effectuées visent à rendre le système émulé « au courant » du fait qu'il s'exécute dans une machine virtuelle. De ce fait, il pourra collaborer plus étroitement avec le système hôte, en utilisant une interface spécifique, au lieu d'accéder au matériel virtuel via les couches d'abstraction. Au final, l'architecture obtenue est plus performante que l'empilement de couches d'abstraction.

En pratique, un système paravirtualisé possède quelques pilotes de périphériques et sous-systèmes modifiés, qui lui permettent de communiquer directement avec la machine virtuelle, sans avoir passé par une couche d'abstraction pour parler au matériel virtuel. Les pilotes paravirtualisés échangent directement des données avec la machine virtuelle, sans avoir à passer par une émulation du comportement du matériel.

Les parties du système hôte généralement modifiées pour tirer profit de la paravirtualisation sont la gestion de la mémoire et la gestion des E/S. En effet, ce sont véritablement les deux goulets d'étranglement d'un système virtualisé, du fait du nombre de couches d'abstraction à traverser.

⁴ Jean-François Apréa, Hyper-V (version 3) et SC Virtual Machine Manager, Technologie de virtualisation sous Windows Server 2012

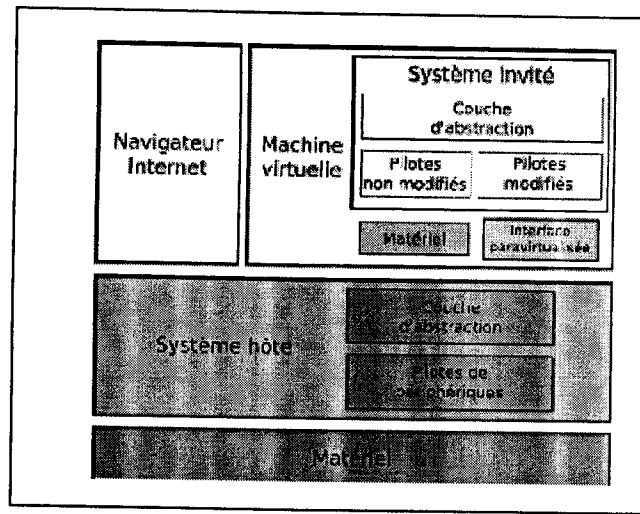


Figure 1.3 : para-virtualisation

La paravirtualisation garde une séparation nette entre le système invité et le système hôte. De ce fait, seul le système hôte a un accès direct et exclusif au matériel. Le système invité doit donc toujours passer par la machine virtuelle pour accéder au matériel, qui passe à son tour par la couche d'abstraction. On peut donc améliorer davantage le processus en laissant au système invité un accès direct mais contrôlé au matériel. C'est le but des systèmes à hyperviseur .

6.3. Le Système a Hyperviseur

Un « hyperviseur »⁵ est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une machine physique en même temps. L'Hyperviseur est un noyau hôte allégé et optimisé pour ne faire tourner que des noyaux des systèmes invités adaptés et optimisés pour tourner sur cette architecture spécifique, les systèmes invité ayant conscience d'être virtualisés.

L'utilisation d'un hyperviseur est en quelque sorte l'évolution logique de la paravirtualisation, l'on recherche encore une amélioration des performances. Dans les technologies précédentes, le système hôte était le seul à avoir un accès direct au matériel ; avec hyperviseur, le système hôte partage cet accès avec les systèmes invités.

On parle de portage, de la même manière qu'on porte un système ou une application vers une nouvelle architecture matérielle.

⁵ ibid

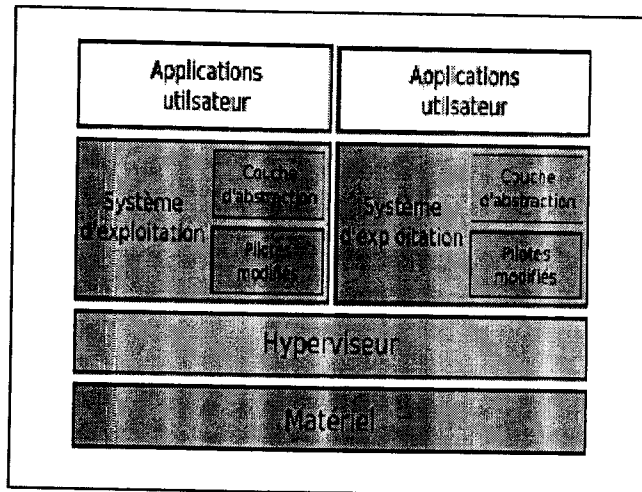


Figure 1.4 : hyperviseur

Au démarrage de l'ordinateur, c'est normalement le système d'exploitation qui prend la main et contrôle le matériel. Dans le cas de l'utilisation d'un hyperviseur, c'est un système minimaliste, l'hyperviseur, qui prend le contrôle du matériel. Ensuite, il fait appel à un système d'exploitation complet, qui sera donc exécuté par-dessus l'hyperviseur.

Ainsi, le système d'exploitation doit passer par l'hyperviseur pour tout accès au matériel. On peut donc très facilement installer un deuxième système d'exploitation, qui passera lui aussi par l'hyperviseur pour l'accès au matériel. Comme les systèmes d'exploitation doivent obligatoirement passer par ce dernier pour tout accès au matériel, l'hyperviseur peut assurer qu'ils n'accèdent qu'aux ressources autorisées, sans perturber le fonctionnement des autres systèmes.

A la différence des deux technologies vues précédemment, la virtualisation complète et la paravirtualisation, l'hyperviseur est le seul à un accès privilégié au matériel. Il n'y a cette fois pas d'accès direct au matériel pour le système d'exploitation, uniquement une couche d'abstraction minimale fournie par l'hyperviseur.

Les systèmes à hyperviseur contrôlent de façon plus fine l'accès au matériel et l'utilisation des ressources. Au niveau de l'hyperviseur, on veillera donc qu'un système ne dérange pas les autres en consommant trop de ressources, alors que dans le système de virtualisation complète, on s'assurera qu'un programme utilisateur ne dérange pas la machine virtuelle qui exécute les autres systèmes d'exploitation en tant que systèmes invités. Tous les systèmes destinés à s'exécuter au-dessus d'un hyperviseur doivent être

portés, comme les systèmes invités pour la paravirtualisation. Cette opération vise à adapter les couches basses du niveau système d'exploitation pour qu'elles communiquent avec l'hyperviseur plutôt qu'avec le matériel.

7. Logiciel de virtualisation : Virtualbox

Nous sommes intéressés dans notre projet par l'application du virtualisation virtualbox. C'est Un logiciel formidable et techniquement avancé, Virtualbox, qui vous permettra de faire tourner un système dans un autre, tout cela de manière « virtuelle ». Découvrons d'abord ensemble ce qu'est VirtualBox avant de nous intéresser à son installation et à son fonctionnement. Un logiciel qui permet au débutant et à l'utilisateur expérimenté, de découvrir une distribution.

Bien entendu, seul le système hôte pourra interagir réellement avec les périphériques. Pour pouvoir faire tourner VirtualBox sur une machine, il faudra que celle-ci soit assez solide avec un processeur pas trop lent, suffisamment de Ram et de l'espace libre (de la place sera prise sur le disque dur physique pour créer l'espace virtuel). En gros, on pourra dire qu'il vous faudra en moyenne un processeur tournant entre 1 Ghz et 2 Ghz minimum, 1 Go de mémoire vive et un espace disque d'environ 10 Go pour une utilisation confortable.

Sous VirtualBox, la manipulation des machines virtuelles nécessite plusieurs étapes :

- ✓ **création d'un disque dur virtuel (VDI).** Nous pouvons soit créer un disque virtuel de taille fixe, soit utiliser un live-cd. Cette deuxième solution permet d'obtenir un disque virtuel n'occupant que peu de place ;
- ✓ **créer une nouvelle machine virtuelle.** Un fichier de description de la machine, comportant des différents paramètres de configuration, est créé ;
- ✓ **rattacher le disque VDI et l'image ISO** du système d'exploitation invité à la machine virtuelle ;
- ✓ **configurer le réseau** de la machine virtuelle
- ✓ **Lancer la machine virtuelle.**

7.1. Configuration réseau de virtualbox:

Virtualbox possède 3 modes de fonctionnement réseau :

- **Réseau interne** : les machines virtuelles sont confinées entre elles dans un réseau virtuel.
- **NAT** : Les machines virtuelles peuvent accéder à internet via une connexion NAT.
- **Adaptateur réseau hôte** : Les machines virtuelles utilisent une interface de l'ordinateur hôte pour accéder au réseau.

Exemple :

Sur notre ordinateur hôte, nous pouvons voir des machines virtuelles (réseau 10.0.X.X). Ces machines virtuelles sont connectées sur des interfaces virtuelles appelées Tap. Les Tap sont elles-même connectées sur un pont virtuel « (bridge). »⁶

Le Bridge virtuel joue le rôle de switch et pont réseau, via ce bridge les interfaces Tap peuvent communiquer entre elles. Mais il est possible d'y attacher aussi une interface physique de l'ordinateur hôte (eth0). Dans le but d'atteindre un autre ordinateur physique qui possède lui aussi des machines virtuelles.

Il est donc possible de créer un réseau de machines virtuelles reliées par un réseau physique. Cette technique peut être utile si nous voulons ajouter plusieurs machines virtuelles sur des ordinateurs qui n'en supportent qu'un nombre limité.

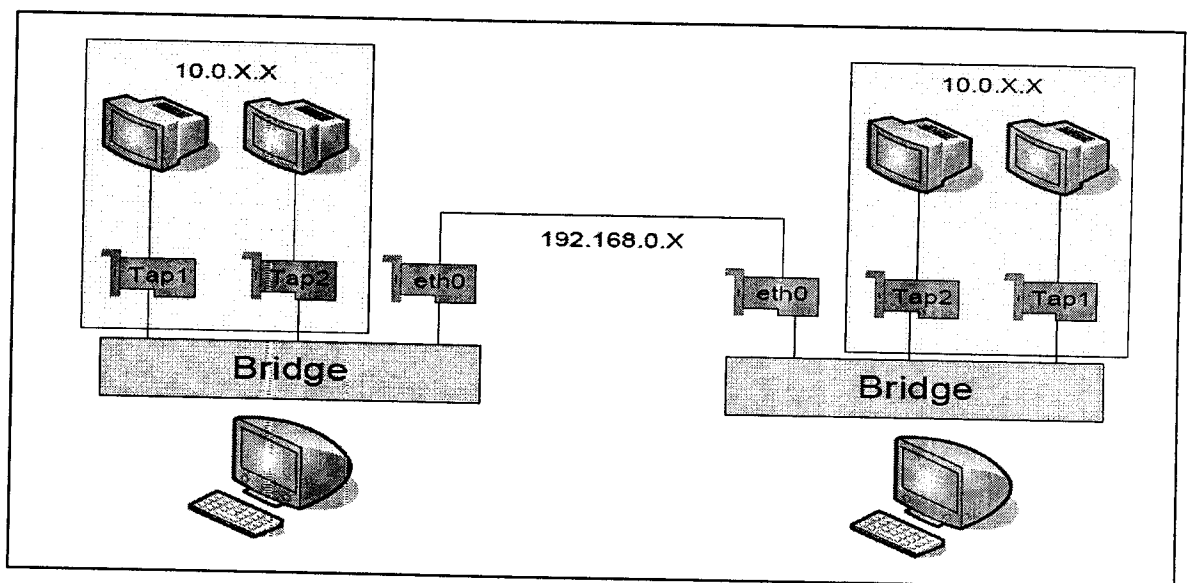


Figure 1.5 : réseau virtuel

⁶ Tunisien team ; Wajih Letaief Virtualisation sous Ubuntu avec VirtualBox novembre 2008

8. Conclusion

Dans ce chapitre nous venons de voir la virtualisation et son importance, les différents types de virtualisations et leurs applications ainsi que les différents acteurs de la virtualisation.

Dans le chapitre suivant, nous présenterons les étapes d'installation et configuration d'un serveur dns, un serveur web apache et un serveur de transfert de fichier vsftp.

Chapitre N°2

Administration d'un serveur web et un serveur ftp

1. Introduction

L'architecture client-serveur désigne un mode de communication entre des ordinateurs ou des logiciels. Les mots « serveur » et « client » peuvent soit désigner :

Les ordinateurs, on parle alors de serveur informatique et de poste client, soit désigner les logiciels fonctionnant sur ces ordinateurs, on parle alors de logiciel serveur [ou service] ou de logiciel client.

Le serveur est à l'écoute sur un réseau informatique, prêt à répondre aux requêtes envoyées par des clients.

Les clients sont pilotés par les utilisateurs et envoient des requêtes au serveur, puis attendent la réponse pour la donner à l'utilisateur. un serveur est capable de servir plusieurs clients simultanément.

Dans ce chapitre, nous présentons les étapes nécessaires pour monter un serveur web apache et un serveur de transfert de fichier vsftp (very secure ftp) sous linux en utilisant la distribution ubuntu 9.10.

2. Serveur WEB

Un serveur Web est un ordinateur connecté en permanence à l'Internet et sur lequel fonctionne un logiciel appelé " serveur ".

N'importe quel ordinateur connecté à l'Internet peut envoyer une requête au serveur web et lui demander de lui transmettre une page html et tous les éléments qui y sont indiqués.

Le serveur répond en transmettant les informations demandées. s'il les possède.

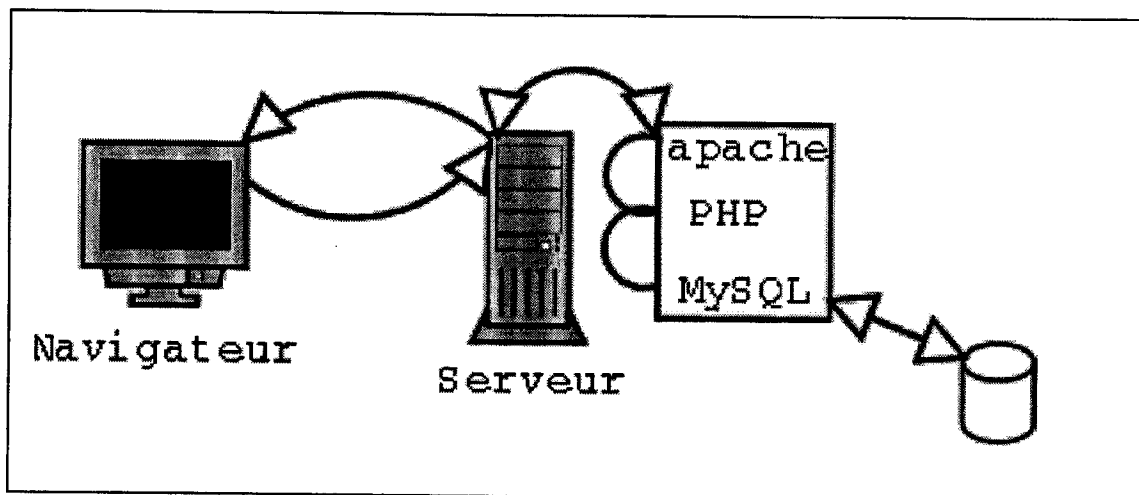


Figure 2.1 : serveur web

L'image ci-contre présente une pile de serveurs web chez un hébergeur. Ce sont souvent des ordinateurs un peu différents de ceux que nous utilisons quotidiennement.

Souvent, les particuliers utilisent l'espace web offert par leur fournisseur d'accès à l'Internet ou des serveurs web gratuits qui offrent leur service en échange d'une publicité apposée sur les pages du site.

Si l'on souhaite plus de fiabilité, on préférera "acheter" de l'espace disque sur un serveur Web commercial. Un grand nombre de sociétés offrent ce service pour des prix très variables.

Les ordinateurs sur le web se connecteront donc au serveur chez notre hébergeur pour consulter nos sites.

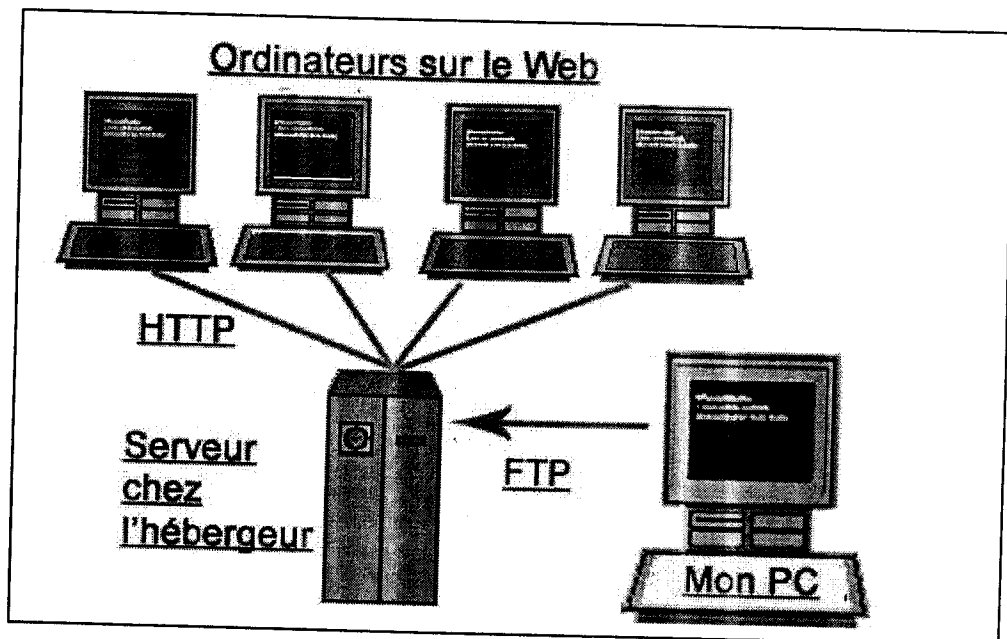


Figure 2.2 : hébergement d'un serveur web

Pour envoyer nos pages sur le serveur Web de notre hébergeur, il nous faut utiliser le protocole **FTP** (file transfer protocol).

Synthèse:

Les principaux acteurs qui ont été décrits sur cette page sont :

- notre ordinateur sur lequel sont enregistrées les pages de notre site
- le serveur chez l'hébergeur de notre site
- les ordinateurs des internautes visitent notre site.

La communication entre ces ordinateurs se fait :

- par le protocole **FTP** entre notre ordinateur et le serveur chez notre hébergeur
- par le protocole **HTTP** du web entre le serveur chez notre hébergeur et les ordinateurs connectés au web.

3. Serveur web apache

- Apache est « un serveur web http libre », c'est un des serveurs web les plus utilisés sur Internet avec plus de 60% des sites d'Internet.
- Un serveur http apache est un serveur hébergeant un ou plusieurs sites Web c'est à dire des pages html ou des programmes générant des pages html qui sont Accessibles par des navigateurs internet (client web).
- Le protocole, permettant l'échange de pages html est le protocole http, d'ou le nom de serveur http. Ce protocole utilise généralement le port 80.
- Apache est un serveur modulaire. Chaque module permet d'ajouter des fonctionnalités au serveur.
- Apache peut gérer plusieurs sites web en même temps, ayant chacun leur nom, à l'aide des hôtes virtuels

3.1. Le protocole http utilisé par web :

HTTP ou HyperText Transfer Protocol est un protocole de requêtes et de réponses.

Le dialogue entre un client web (un navigateur internet) et un serveur (Apache) se traduit par une requête du client à laquelle le serveur répond en effectuant le traitement intermédiaire adéquat.

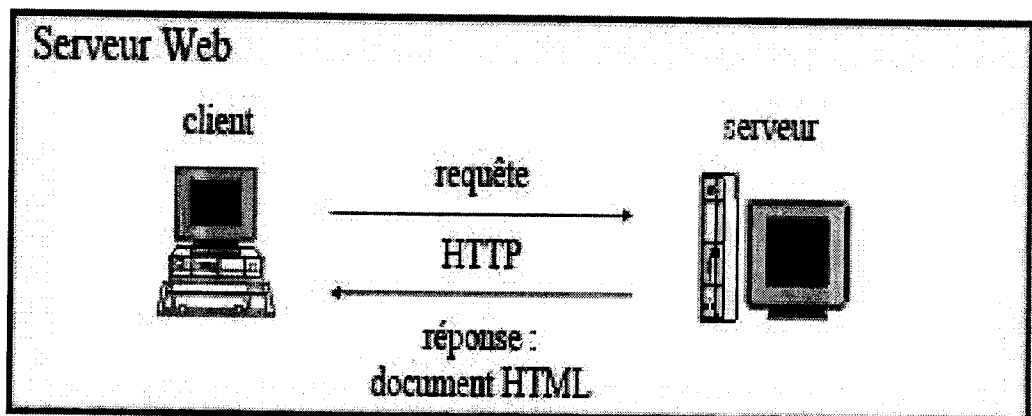


Figure 2.3 : schéma d'un serveur web

3.2. Installation et configuration apache

Les étapes d'installation d'un serveur web apache2 :

Pour installer le serveur web apache on doit lancer la commande suivante :

➤ sudo apt-get install apache2

démarrage de serveur web

➤ sudo /etc/init.d/apache2 start

stopper le serveur web :

➤ sudo /etc/init.d/apache2 stop

relancer le serveur web

➤ sudo /etc/init.d/apache2 restart

le chemin des fichiers de configuration

➤ /etc/apache2 : contient un fichier de configuration apache2.conf

Le site web par défaut se trouve dans le chemin par défaut : /var/www/(en trouve le site index.html).

Authentification aux sites hébergés

Pour un accès authentifié nous utilisons deux fichiers : .htpasswd et .htaccess

Le fichier .htpasswd contient les utilisateurs qui ont le droit de consulter la page web

Le fichier .htpasswd se trouve dans le chemin /etc/apache2 (c'est un fichier caché)

Pour créer un nouvel utilisateur on utilise la commande suivante :

htpasswd -c .htpasswd nom-utilisateur

Pour le fichier .htaccess sera placé dans le répertoire ou se trouve les pages web a protégées

Il contient les informations suivantes :

Authname "sécurité web"

AuthUserFile /etc/apache2/.htpasswd

AuthType Basic

require valid-user |list-user

Fichier /etc/apache2/site-enable/default contient les informations sur le chemin ou se trouve les sites web a consultées pas les clients de navigateur Internet

Créer un site web appelé index.html

Créer un répertoire web dans le chemin /home/licence-pfe/

Mettre le site web index.html dans le répertoire web (/home/licence-pfe/web)

Modifier le fichier : /etc/apache2/site-enable/default est ajouter les lignes enfoncés:(en gras)

Exemple 1

```

<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    Alias /essai/ "/home/licence-pfe/web/"
    <Directory "/home/licence-pfe/web/">
        Options Indexes FollowSymLinks MultiViews

        AllowOverride AuthConfig
        Order deny,allow
        deny from all
        Allow from 192.168.2.0
    </Directory>
</VirtualHost >

```

Le site web a hébergé se trouve dans le chemin /home/licence-pfe/web avec un raccourci indiqué par /essai/

Pour accéder au site web on écrit soit sous linux ou Windows a partir d'un navigateur Internet sur une machine cliente :

<http://192.168.2.2/essai/index.html>

4. Serveur ftp

Le File Transfer Protocol (protocole de transfert de fichiers), ou FTP, est un protocole de communication destiné à l'échange de fichiers sur un réseau TCP/IP.

Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.

La variante de FTP protégée par SSL s'appelle FTPS.

FTP obéit à un modèle **client/serveur**, c'est-à-dire qu'une des deux parties, le *client*, envoie des requêtes auxquelles réagit l'autre, appelé *serveur*.

En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé *serveur FTP*, qui rend publiquement une arborescence de fichiers similaire à un *système de fichiers Unix*

Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique ou en ligne de commande)

Client en ligne de commande

ftp, Wget, Curl

Client avec interface graphique

CuteFTP, FileZilla et KamzyFTP (windows)

Gftp

FireFTP extension pour Firefox

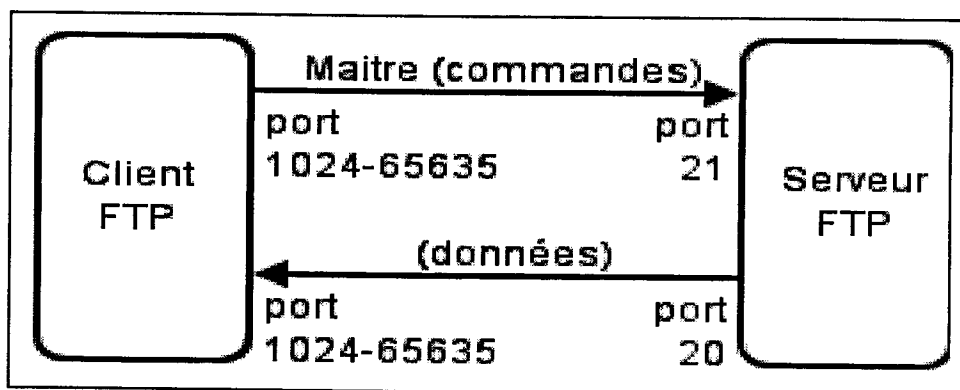


Figure 2.4 : serveur FTP

4.1. Protocole FTP

Le protocole utilise deux types de connexions TCP :

Une connexion de *contrôle* initialisée par le client, vers le serveur (port 21 en général), pour transmettre les commandes de fichiers (transfert, suppression de fichiers, renommage, liste des fichiers).

Une connexion de *données* initialisée par le client ou le serveur pour transférer les données requises (contenu des fichiers, liste de fichiers).

4.2. Mode de fonctionnement d'un serveur ftp

Mode ftp actif

C'est le mode par défaut des clients FTP. Le client établit dans un premier temps une session TCP sur le port 21 (FTP) du serveur ("control channel"). Une fois la session établie et l'authentification FTP acceptée, c'est le serveur qui établit une session TCP (avec le port source 20, FTP-DATA) vers un port dynamique du client ("data channel").

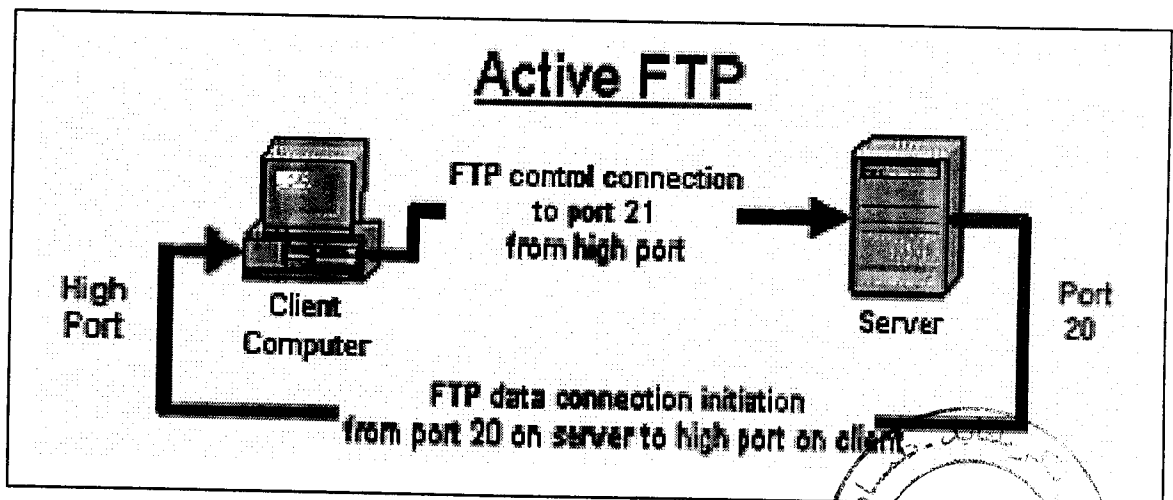


Figure 2.5 : mode ftp actif

Mode ftp passif

Comme pour le FTP actif, « le client établit une première session TCP sur le port 21(FTP)⁷ du serveur ("control channel")⁸. Une fois la session établie et l'authentification FTP acceptée, on demande au serveur de se mettre en attente de

⁷Laurent Bloch et Christophe Wolfhugel, « Sécurité Informatique : Principes et méthode », Juin 2011.

⁸Liran Lerman, « Mise en place d'un proxy Squid avec authentification Active Directory », Thèse de doctorat, Université limôge, 2008

session TCP grâce à la commande PASV. Alors que le client peut établir une seconde session TCP sur un port dynamique vers le serveur ("data channel").

Le numéro de port dynamique est transmis du serveur vers le client suite à la commande PASV.

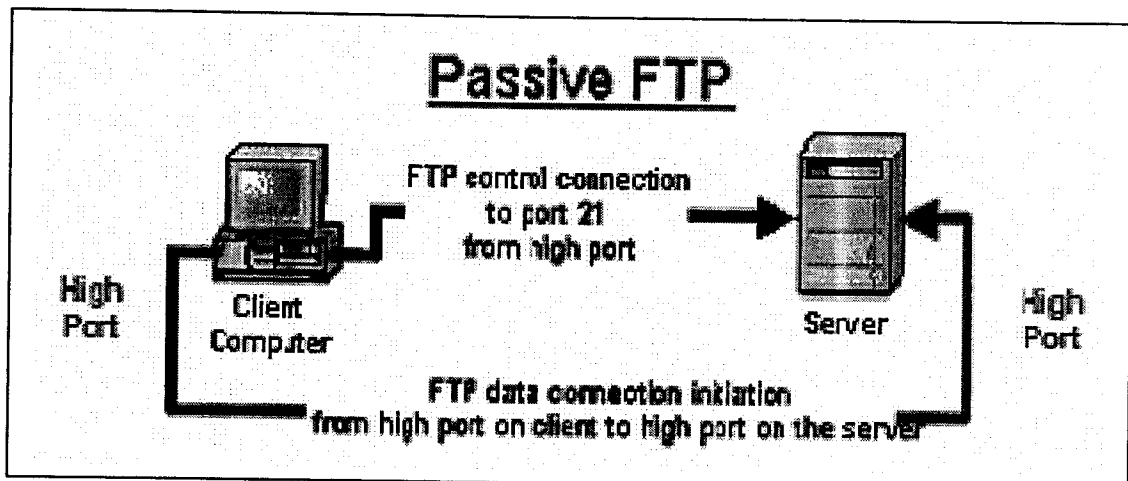


Figure 2.6: mode ftp passif

4.3. Configuration d'un serveur vsftpd sous linux

Voilà les différentes étapes pour installer un serveur ftp

Installation du paquet vsftpd :

```
$apt-get install vsftpd.deb
```

Démarrer le serveur FTP

```
sudo /etc/init.d/vsftpd restart
```

Arrêter le serveur FTP

```
sudo /etc/init.d/vsftpd stop
```

Exemple de fichier vsftpd.conf

Voici un exemple de configuration plus complexe, qui permet d'autoriser les comptes utilisateurs présents sur le serveur à se connecter à leurs dossiers personnels, sans autoriser l'accès anonyme :

On indique la bannière

```
ftpd_banner = Bienvenue sur le serveur licence-pfe
```

Le serveur doit-il fonctionner en mode standalone (autonome)

```
listen=YES
```

```
# On indique le port d'écoute tcp du serveur, par défaut 21
#listen_port=6996
# interdire les connexions anonymes (Valeur = NO)
anonymous_enable=NO
# On interdit l'écriture anonyme
anon_upload_enable=NO
# On interdit la création de répertoires anonyme
anon_mkdir_write_enable=NO
# On interdit la création, suppression, et le renommage de répertoire
anon_other_write_enable=NO
# Accepte t-on les connexions des utilisateurs locaux
local_enable=YES
# Accepte t-on l'écriture de fichier (commandes STOR, DELE, RNFR, RNTD,
MKD, RMD, APPE et SITE)
write_enable=YES
# On indique que tout les utilisateurs sont limités à leurs propres répertoires
chroot_local_user=YES
chroot_list_enable=NO
# On fixe le masque local à 022 (les fichiers écrits auront les droits 755)
local_umask=022
# On active le log des actions des utilisateurs
xferlog_enable=YES
# Indique le chemin du fichier de log
xferlog_file=/var/log/vsftpd.log
# On vérifie que la commande PORT provienne bien du port 20 de la machine
cliente
connect_from_port_20=YES
# On indique les valeurs des timeout
idle_session_timeout=300
data_connection_timeout=120
connect_timeout=60
accept_timeout=60
```

On interdit la commande ABOR

async_abor_enable=NO

On interdit les transferts ASCII

ascii_upload_enable=NO

ascii_download_enable=NO

L'heure locale sera utilisée pour l'enregistrement des fichiers

use_localtime=YES

Pour connecter à un serveur ftp on utiliser firefox (navigateur internet) ou client ftp :

ftp://@ip serveur_ftp

Exemple : **ftp://192.168.2.2/**

192.168.2.2 adresses ip de serveur ftp

5. Conclusion

Dans ce chapitre, nous avons présentés les différentes étapes d'installation d'un serveur web et d'un serveur ftp sur notre machine virtuelle. L'objectif de ces deux services est de contrôler l'accès de notre client web de la machine virtuelle Windows par notre serveur proxy qui est installé sur une machine virtuelle routeur.

Dans le prochain chapitre, nous présenterons les étapes d'installation et de configuration d'un serveur proxy squid et squidgard.

Chapitre N°3

Configuration d'un serveur proxy squid

1. Introduction

Dans un réseau local, on peut avoir envie de mettre une machine qui fasse l'intermédiaire entre notre réseau et Internet. Le serveur proxy « (ou serveur mandataire) »⁹ permet de d'envoyer les requêtes et de recevoir les réponses à la place de ses clients. Ces requêtes peuvent être des requêtes de divers protocoles, les plus utilisées étant le HTTP, HTTPS et FTP, SSL.

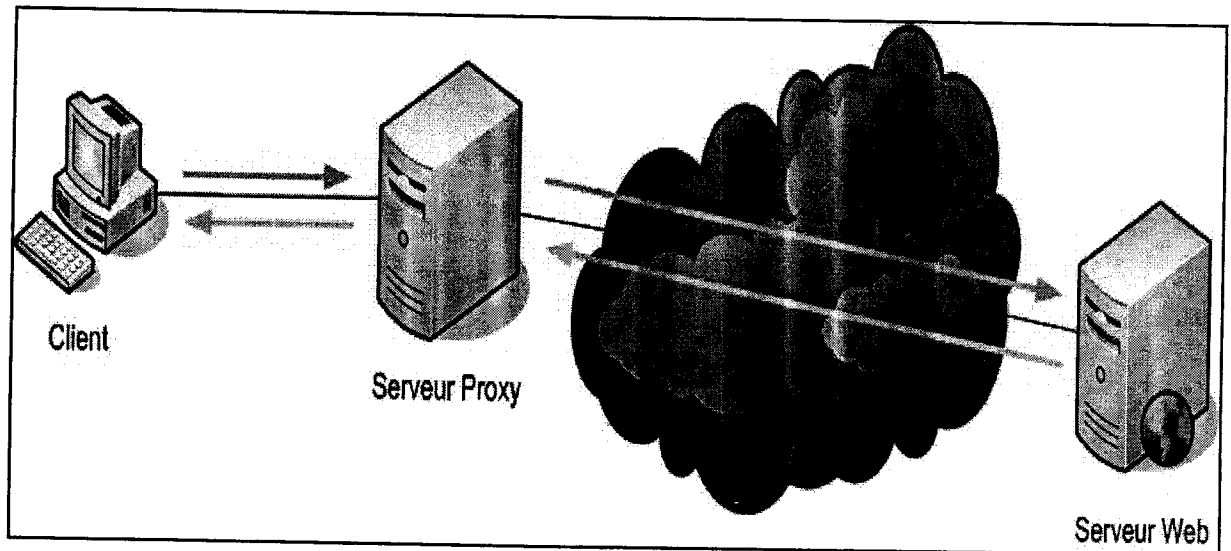


Figure 3.1 : serveur proxy

Les clients deviennent invisibles pour l'internet. Si un client est derrière un proxy, les autres machines sur internet penseront qu'il s'agit du serveur proxy.

2. Les avantages d'un proxy

2.1. Utilisation d'un cache

Le cache permet de stocker un certain nombre de fichiers pendant que vous naviguez sur Internet pour permettre d'afficher la page plus rapidement si vous retournez sur le même site une autre fois.

En général, les navigateurs Web utilisent un cache interne moins volumineux. Un serveur cache-proxy permet de faire la même chose à un plus grand niveau: Il est dédié au stockage des fichiers et pages Internet les plus visitées.

⁹ Loïc Thomas, Squid, Intégrez un proxy à votre réseau d'entreprise, édition , Eni 2012

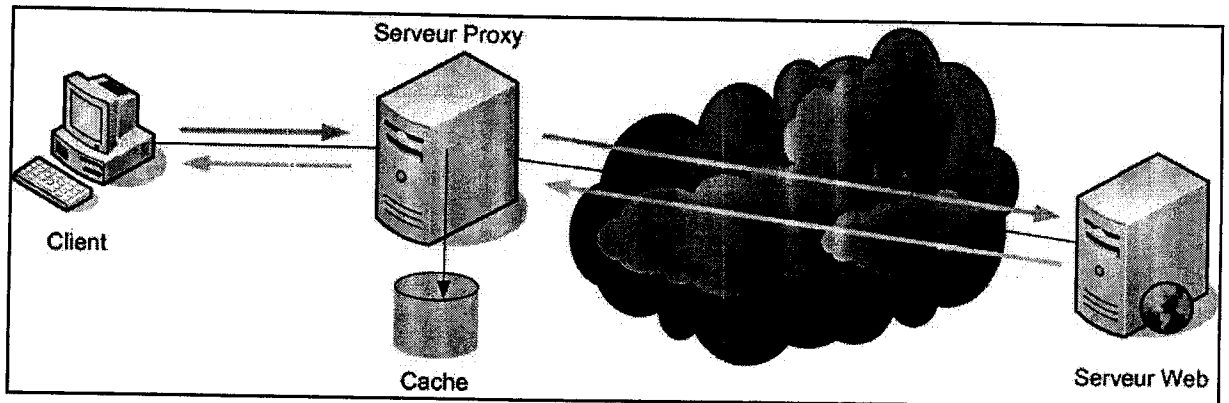


Figure 3.2 : cache d'un serveur proxy

2.2. Filtrage

Le serveur proxy peut également servir à suivre toutes les entrées et sorties en créant des journaux d'activités « (logs) »¹⁰ qui enregistrent chaque requête que font les clients.

Au niveau des clients on peut lister un certain nombre de sites autorisés (liste blanche) ou des sites qui ne le sont pas (liste noire).

Au niveau des serveurs, l'analyse des réponses en fonction de certains critères s'appelle le filtrage de contenu (mots clés, adresses IP, noms de domaines, ...).

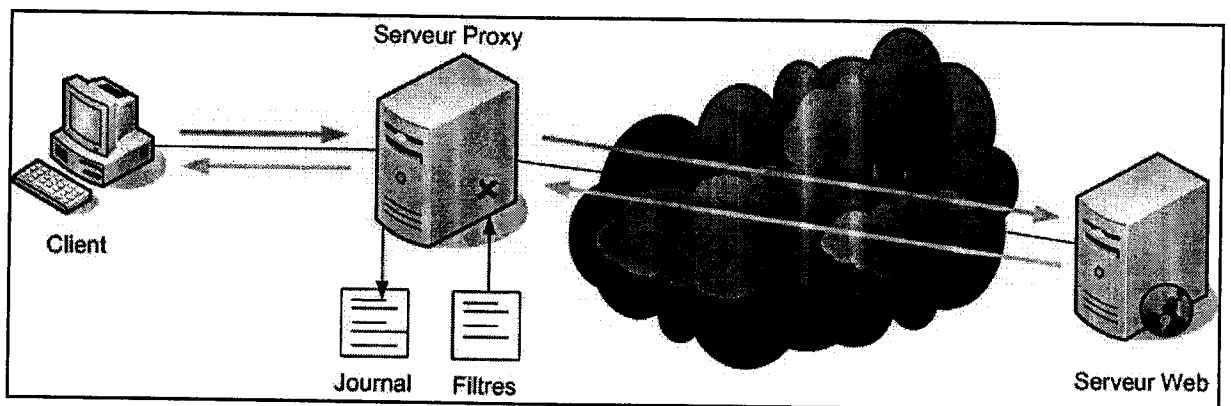


Figure 3.3 : filtrage d'un serveur proxy

2.3. Authentification client

Puisque le serveur Proxy se trouve entre le réseau local et Internet, il peut imposer que le client « s'authentifie pour se connecter » (login et mot de passe).

¹⁰ Kulbir Saini, Squid Proxy Server 3.1 Beginner's Guide.

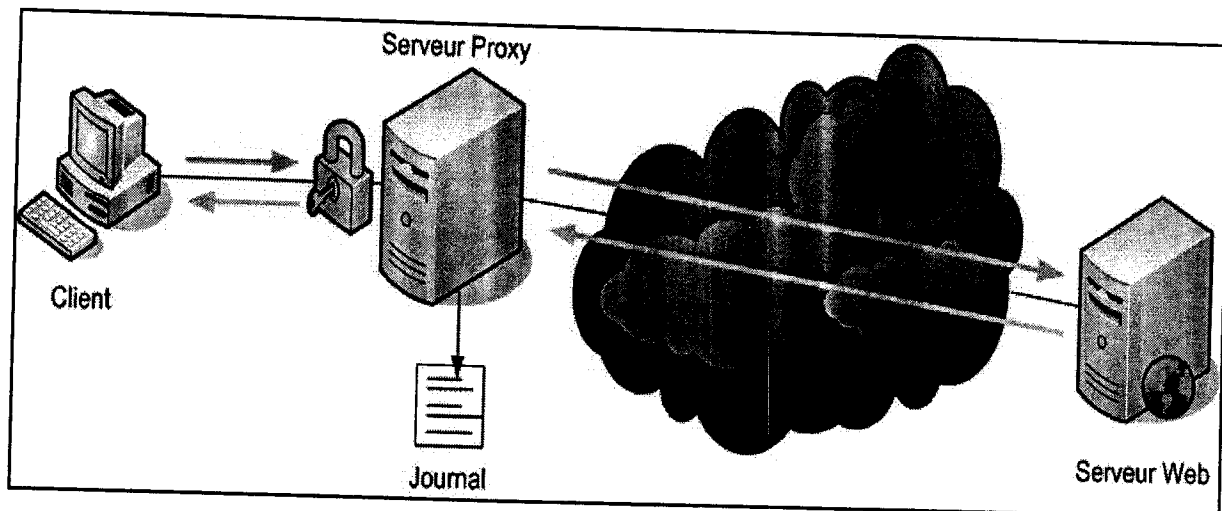


Figure 3.4 : authentification par un serveur proxy

2.4. Reverse proxy

« Le proxy inversé ne sert pas de relais aux clients de son LAN »¹¹ mais à ceux qui viennent d'Internet et qui ont aussi besoin d'avoir un accès à certains serveurs internes.

3. Limitations

Squid n'est pas un Firewall. Il peut limiter les possibilités des clients mais il ne protège pas l'accès aux personnes extérieures au réseau.

La gestion du pare feu sous Linux est réalisée par ipchains pour les noyau 2.2 ou iptables pour les noyau 2.4.

Squid ne supporte pas tous les protocoles. Il est ainsi incapable de gérer les protocoles de news et de vidéo conférences. Il est nécessaire d'utiliser d'autres caches applicatifs.

4. Protocoles supportés

Squid supporte les protocoles suivants :

- HTTP: le protocole Web
- FTP : le protocole de transfert de fichiers
- Gopher : un protocole proche de http inusité aujourd'hui
- SSL : utilisé pour les transactions sécurisées
- WAIS : protocole inusité
- ICP : un protocole généraliste permettant la communication entre serveurs de cache.
- HCTP : un protocole de cache orienté http

¹¹ Article > GNU/Linux : Serveur mandataire.

5. Installation et configuration de SQUID

Squid est un proxy assez performant. De base il permet déjà beaucoup de choses: fermer des ports un par un, empêcher des plages d'ip de se connecter à internet via ce proxy.

Mais ce n'est qu'avec le plugin Squidguard qu'il permet de filtrer une à une les URL en fonction des blacklists paramétrables.

Squid peut être couplé avec une interface web (Webmin), rendant ainsi sa configuration beaucoup plus simple que via le .conf qui même si il est bien commenté reste difficile à paramétrer.

Paquets/Installation

Dans le gestionnaire de paquets Synaptic, il faut télécharger les paquets suivants :

- Squid
- squid-common
- squid-langpack

Une fois Squid installé le dossier contenant le .conf de squid et le dossier contenant le .conf de squidguard qui est situé dans le chemin suivant : /etc/squid

Commandes d'utilisation de squid

Voici les lignes de commandes utiles de Squid :

/etc/init.d/squid start Lance Squid

/etc/init.d/squid stop Stoppe Squid

/etc/init.d/squid restart Stoppe et relance Squid

/etc/init.d/squid status Donne le status de Squid : Lancé ou non

/etc/init.d/squid reload Recharge les .conf de Squid si il est déjà lancé (reload le .conf de squidguard également)

5.1. Administration et configuration d'un squid

Nommer le proxy

Pour nommer un squid on utilise le paramètre `visible_hostname`

visible_hostname nom_de_votre_pc

Choisir le port

Le serveur proxy étant nommé, nous allons entrer un peu plus dans la configuration de squid en choisissant le port qui sera en écoute. Par défaut, celui ci est 3128

http_port 3128

Répertoire de cache

Indique ou seront stockés les objets « cachées ». Ce répertoire doit appartenir à l'utilisateur définit dans la directive « cache_effective ». Il faut également indiquer le type d'organisation de fichier et l'espace disque alloué.

Voici la syntaxe :

cache_dir <type disk> <répertoire> <taille en Mo> <nb rép><nb sous rép>

En générale, le type est ufs et il faut créer ces répertoires avant l'utilisation par la commande squid -z.

cache_dir ufs /www-cache 1024 16 256

Adresse administrateur

Indique l'adresse mail de l'administrateur du programme. En cas de plantage par exemple un courrier lui sera renvoyé.

cache_mgr admin@licence_pfe.dz

Taille maximale des objets

Squid stocke les objets dans son cache selon une taille définit par des limites inférieures et supérieures.

minimum object size 0 kb

maximum object size 1024 kb

Cache DNS

Squid effectue des requêtes DNS qui sont « bloquantes ». Il démarre ainsi un certains nombre de processus pour répondre à ces requêtes. On spécifie leur nombre avec la directive dns_children. Si cette requête a réussie, elle est placée en cache pendant le temps précisée dans la directive positive_dns. Dans le cas contraire, le temps sera celui de negative_dns.

dns children 10

positive dns ttl 24 hours

negative dns ttl 5 minutes

5.2. Les Contrôles d'accès (ACL)

La sécurité est l'une des fonctionnalités les plus intéressantes de Squid. Celui-ci dispose, en effet, de fonctions de filtres sur des protocoles non Supportés comme http ou ftp par les pare - feu de filtres à paquets.

Il est possible de restreindre l'utilisation de Squid à certaines personnes.

Ces limitations sont écrites à partir d'ACL « (Access Control List) »¹².

« Les ACL sont des règles que le serveur proxy applique. Cela permet par exemple d'autoriser ou d'interdire certaines transactions.

On peut autoriser ou interdire en fonction du domaine, du protocole, de l'adresse IP, du numéro de port, d'un mot, on peut aussi limiter sur des plages horaires.

La syntaxe d'une ACL est la suivante :

acl nom_acl type_acl string

http_access allow|deny [!]nom_acl

type_acl peut prendre comme valeur :

- **src** (pour la source) : indication de l'adresse IP du client sous la forme adresse/masque. On peut aussi donner une plage d'adresses sous la forme adresse_IP_debut - adresse_IP_findst (pour la destination)
- **srcdomain** : Le domaine du client
- **dstdomain** : Le domaine de destination.
- **url_regex** : Une chaîne contenu dans l'URL (on peut utiliser les jokers ou un fichier).
- **urlpath_regex** : Une chaîne comparée avec le chemin de l'URL.
- **proto** : Pour le protocole.
- **Port** : pour le port

¹² Ibid p 112

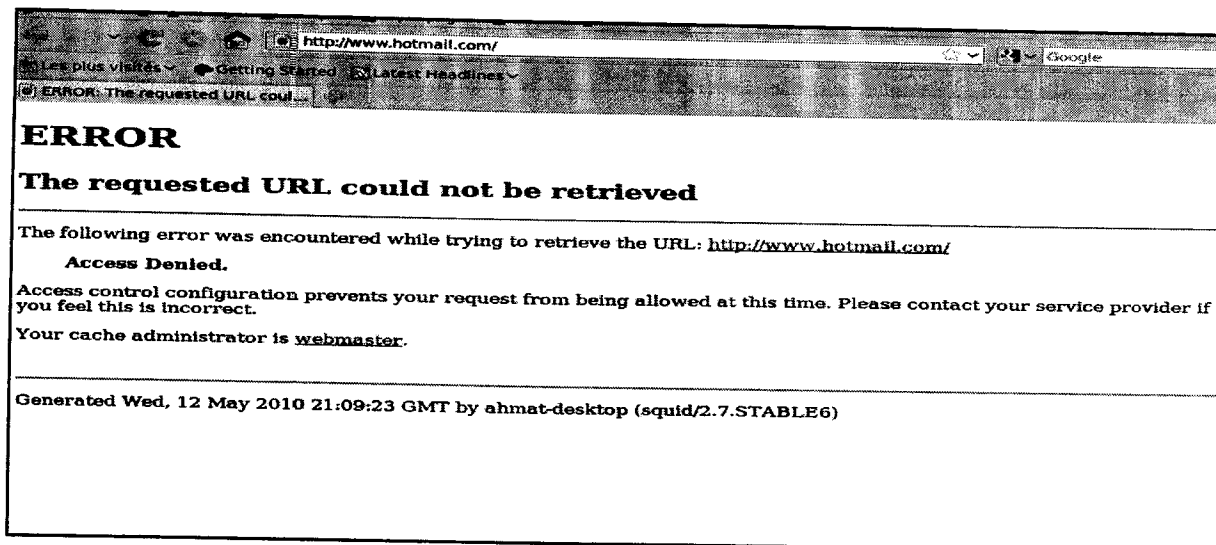


Figure 3.5 : page Access denied

Exemple d'un acl

Un exemple est la gestion du temps des utilisateurs.

Vous pouvez par exemple définir un temps d'utilisation en créant une acl time.

```
acl jour time 08:00-17:30
```

Dans cet exemple, on a défini une journée de 8h00 à 17h30. En dehors de cet horaire, le proxy refusera la connexion.

Mais pour que cela fonctionne, il vous faudra également indiquer à qui cette restriction s'impose.

Exemple, sur l'utilisateur host1 définit tout à l'heure :

```
http_access allow host1 jour
```

Nous avons donc juste rajouté 'jour' à la fin de la ligne d'autorisation créée un peu plus haut.

Ce qui nous donne :

```
#http_access allow our_networks
```

```
http_access allow localhost
```

```
http_access allow host1 jour
```

```
http_access deny all
```

5.3. Installation et configuration de SQUIDGUARD

On installe SquidGuard avec la commande: `apt-get install squidguard`

Une fois ceci fait, on signale à squid d'utiliser squidGuard. Pour cela, on Modifie dans le fichier de configuration `squid.conf` (`/etc/squid/squid.conf`) en ajoutant ce qui suit au début du fichier de configuration :

```
url_rewrite_program /usr/bin/squidGuard
```

```
storeurl_rewrite_children 5
```

5.4. Installation des listes noires

Définition des Blacklists

« Les blacklist ou des listes noires sont des listes d'url ou de noms de domaines que Squid filtrera une fois squidguard configuré et activé. »¹³

Il faut télécharger et compressé la blacklist française qui se trouve dans le fichier `blacklists.tar.gz`

Voila les étapes de création des listes noires

```
tar zxvf blacklists.tar.gz -C /var/lib/squidguard/db/
```

```
cd /var/lib/squidguard/db/
```

```
mv blacklists/* .
```

```
rm -rf blacklists
```

Un autre dossier important est à noter, `/var/lib/squidguard/db`, c'est dans ce dossier que sont rangées les blacklists par défaut.

5.5. Paramétrer SquidGuard

Pour paramétrer Squidguard il n'existe aucune interface graphique comme Webmin pour Squid. Il faut modifier directement `squidguard.conf` dans `/etc/squid` et recharger squid avec `/etc/init.d/squid reload` ou sur webmin.

Voici la configuration par défaut qu'on a utilisé pour le proxy :

¹³] Laurent Bloch et Christophe Wolfhugel, « Sécurité Informatique : Principes et méthode », Juin 2011.

5.6. Configuration squidguard

#Définition des dossiers importants

dbhome /var/lib/squidguard/db

logdir /var/log/squid

#Définition des listes

#Dans le dossier contenant toutes les listes, vous trouverez généralement plusieurs fichiers

#seuls les fichiers domains et urls sont pris en compte par squidguard, il faut les rajouter ici.

```
dest adult {
    domainlist    adult/domains
    urllist       adult/urls
    redirect      http://google.fr
}

dest phishing {
    domainlist    phishing/domains
    urllist       phishing/urls
    redirect      http://google.fr
}
```

#Définition des ACL

```
acl {
    default {
        #Les listes avec un "!" devant sont les listes de liens refusés, celles sans les "!" sont acceptés.
        pass !adult !phishing all

        #Lien vers lequel est redirigé l'utilisateur quand il tent de se connecter sur un des l'un de l'une des
        listes.
        redirect      http://google.fr
    }
}
```

5.7. Compiler les blacklists

Une fois les blacklists ajoutées et paramétrées dans le fichier squidguard.conf, ouvrez un terminal et tapez la commande suivante : `/usr/bin/squidGuard -C all`

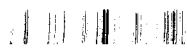
Une fois la compilation terminée, il faut faire un `chmod -R 777 /var/lib/squidguard/db` pour rendre les nouveaux fichiers accessibles à squid. Vous devez ensuite reload squid avec la commande vue plus haut et normalement le filtrage des urls des blacklists devrait être en place.

6. Conclusion

Le squid est l'un des serveurs les plus utilisés par les administrateurs réseaux et les fournisseurs d'accès internet. En outre il faudrait savoir qu'il peut aussi être configuré comme reverse proxy assurant le filtrage en entrée d'un serveur.

Ainsi son utilisation devient plus qu'une nécessité pour tout administrateur soucieux de la sécurité de son réseau.

Dans le prochain chapitre, nous présenterons les étapes d'administration et de test de notre proxy squid en utilisant une interface graphique de l'application Webmin.



Chapitre N°4

Administration proxy squid par webmin

1. Introduction

« Squid est un service serveur proxy-cache sous linux »¹⁵. Les objets consultés par les clients sur internet, sont stockés en cache disque par le serveur. A partir du deuxième accès, la lecture se fera en cache, au lieu d'être réalisée sur le serveur d'origine. De ce fait il permet d'accélérer vos connexions à l'interne en plaçant en cache les documents les plus consultés. On peut aussi utiliser la technique du service serveur mandataire pour effectuer des contrôles d'accès aux sites.

Nous sommes intéressés dans ce chapitre, par l'administration de notre serveur proxy squid en utilisant un gestionnaire graphique des serveurs webmin. Plus nous réalisons les différents tests et scénario de fonctionnement de notre serveur proxy dans un environnement virtuel composé par deux réseaux relié entre par une machine routeur qui va jouer le rôle d'un proxy

2. Architecture de notre réseau virtuel

Nous avons créés 3 machines virtuelles avec deux réseaux à l'aide de virtualbox

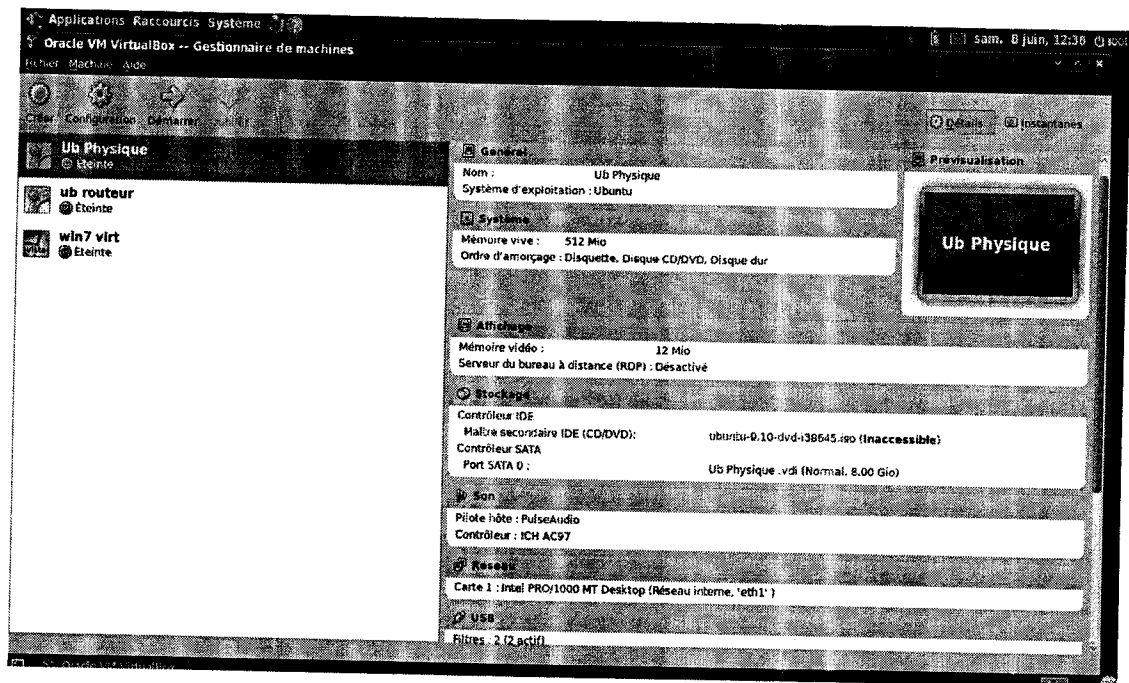


Figure 4.1 : Trois machines virtuelles sous virtualbox

¹⁵ José Dordoigne, Réseaux informatiques, Notions fondamentales et Administration sous Windows ou Linux - Théorie et Pratique - 2 livres en 1.

La première machine est considérée comme une machine serveur web ubuntu 9.10 (serveur apache), représente le réseau externe internet. A la fin de son installation on met au point sa configuration de carte réseau qui devient interne en la nommant « eth1 », et de la même façon on installe la deuxième machine virtuelle Ubuntu 9.10 joue le rôle d'un routeur plus un serveur proxy squid.

Dans la configuration de cette machine on active 2 carte réseau interne la première on la nomme « eth0 » reliée a la 3ème machine, et la deuxième « eth1 » qui relie « le routeur »¹⁶ avec le serveur web ou réseau internet. En fin on crée une dernière machine virtuelle en utilisant le système windows7 qui devient dans notre théorie un client avec les configurations nécessaires (réseau interne « eth0 »).

Voila le schéma de notre réseau virtuel :

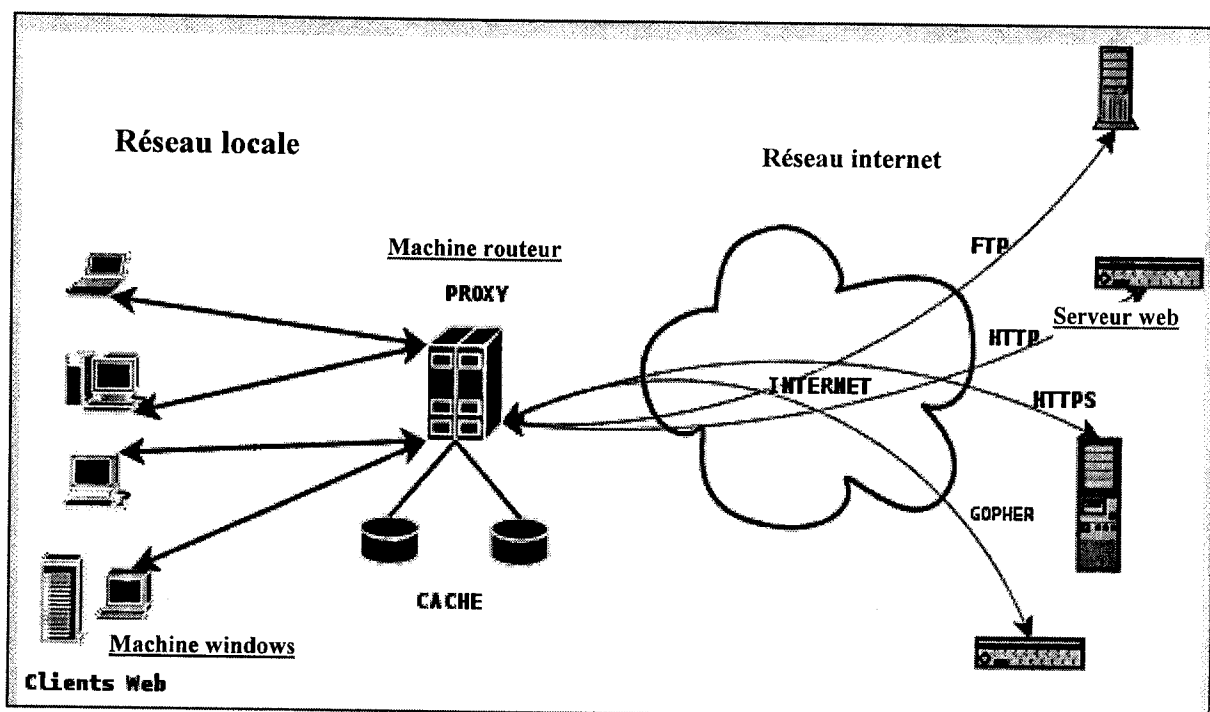


Figure 4.2 : architecture de notre réseau virtuel

Réseau internet possède l'adresse ip : 192.168.2.0

Réseau local possède l'adresse ip 192.168.1.0

La machine Windows possède l'adresse ip 192.168.1.4 (client web) appartient au réseau local

¹⁶ Guy Pujolle, « Les réseaux », Eyrolles, 2008.

La machine serveur web possède l'adresse ip 192.168.2.2 (serveur web apache) appartient au réseau internet.

Le routeur possède deux carte réseaux : @ip de première carte est 192.168.2.1 @ip deuxième carte est 192.168.1.1 c'est une passerelle entre les deux réseaux et en même temps un serveur proxy squid.

Configurer les navigateurs

Voila la configuration de navigateur mozilla

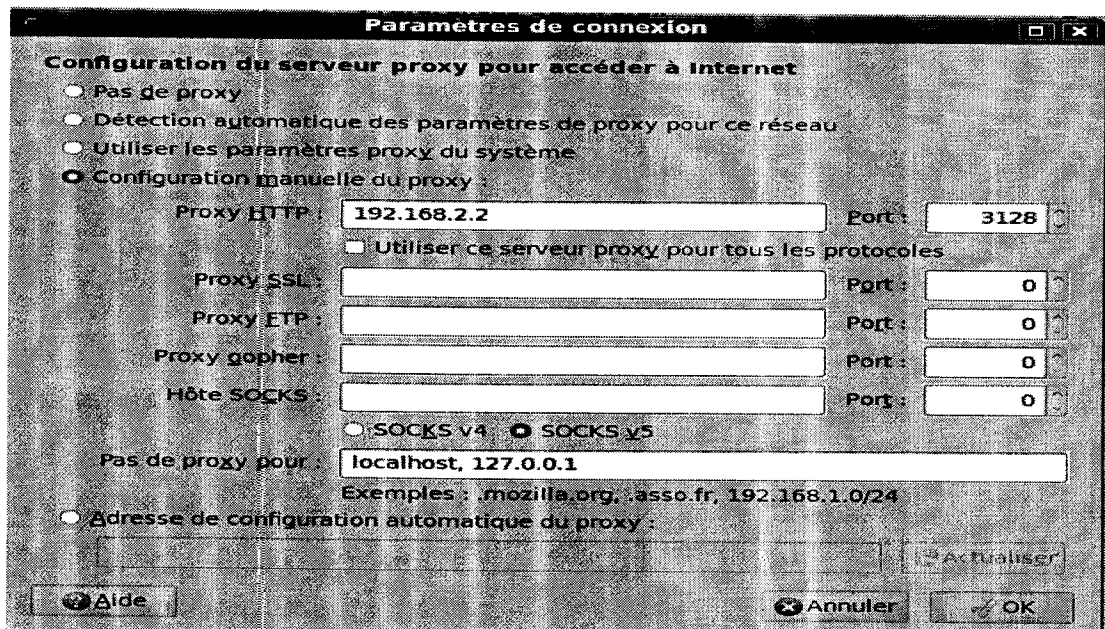


Figure 4.3 : configuration de la connexion à un serveur proxy

3. Administration du serveur proxy squid par webmin

3.1. Présentation de Webmin

Webmin est une application web, ce qui signifie qu'elle doit être utilisée à partir d'un navigateur internet (Firefox, mozilla)

Webmin est un gestionnaire graphique de serveurs. Il n'est pas disponible dans les dépôts linux standards, donc le gestionnaire de paquet Synaptic ne le propose pas. Il faut télécharger le paquet à l'adresse suivante :

<http://sourceforge.net/projects/webadmin/files/webmin/>

Nous avons utilisés la version 1.550 de Webmin sous linux (distribution Ubuntu 9.10).

La configuration de Squid se fera via Webmin.

Après son installation vous pouvez accéder à Webmin via <https://localhost:3128> ou <https://@ip-serveur-proxy:3128>

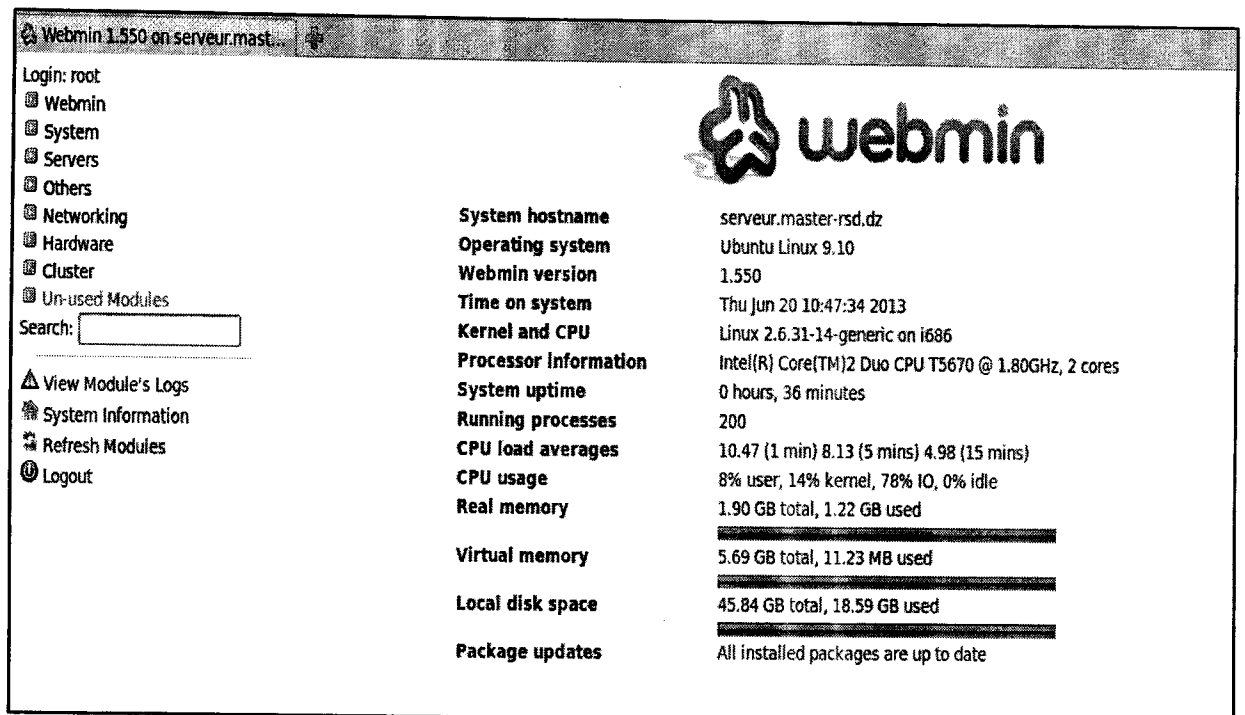


Figure 4.4 : interface graphique de Webmin

Voici la page de démarrage de Webmin qui nous donne les informations sur notre système et sur la version de Webmin (figure 4.4).

On a un aperçu général des fonctionnalités de Squid.

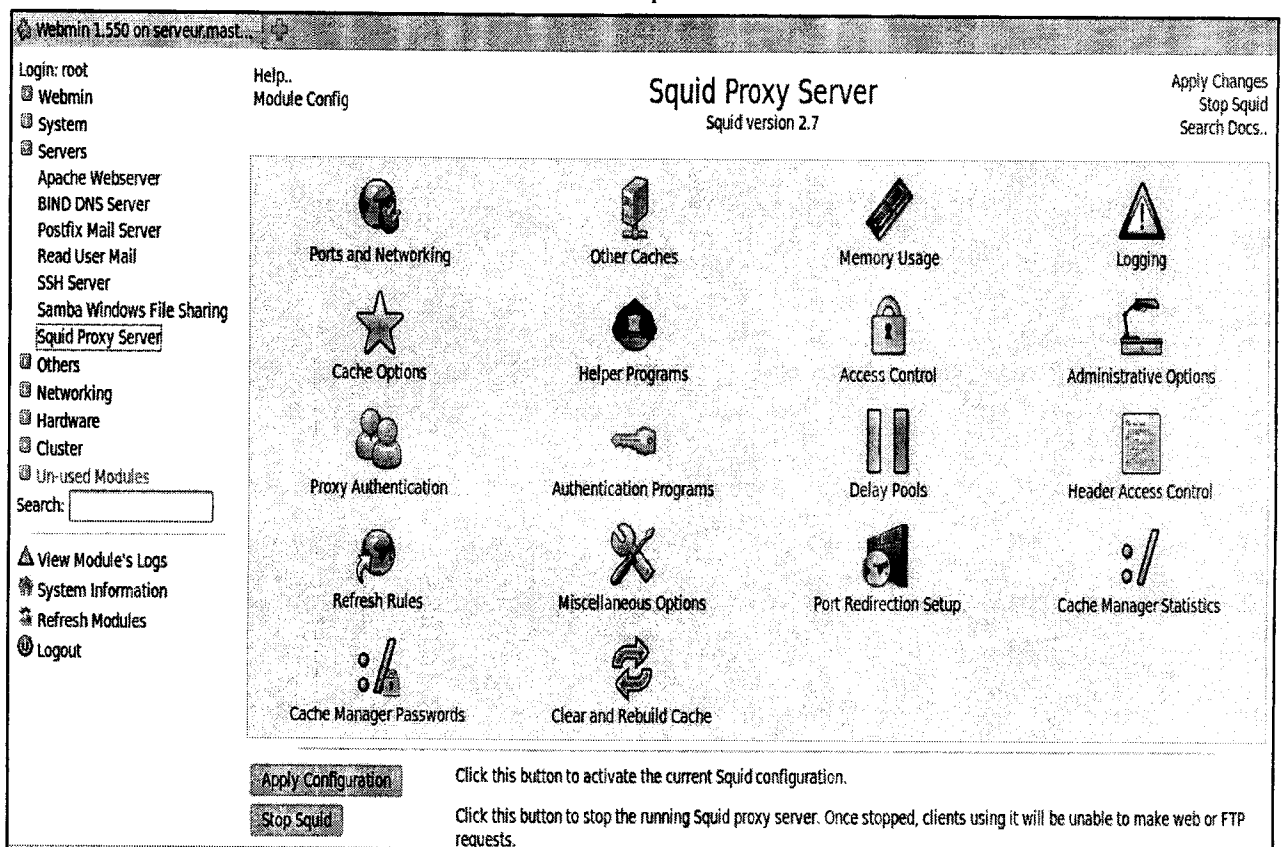


Figure 4.5 : les fonctions de squid sous Webmin

3.2. Configuration des ports et de l'interface d'écoute

Pour ce qui concerne ce serveur la première des choses est de configurer un port et une interface pour pouvoir traiter et renvoyer les requêtes clients.

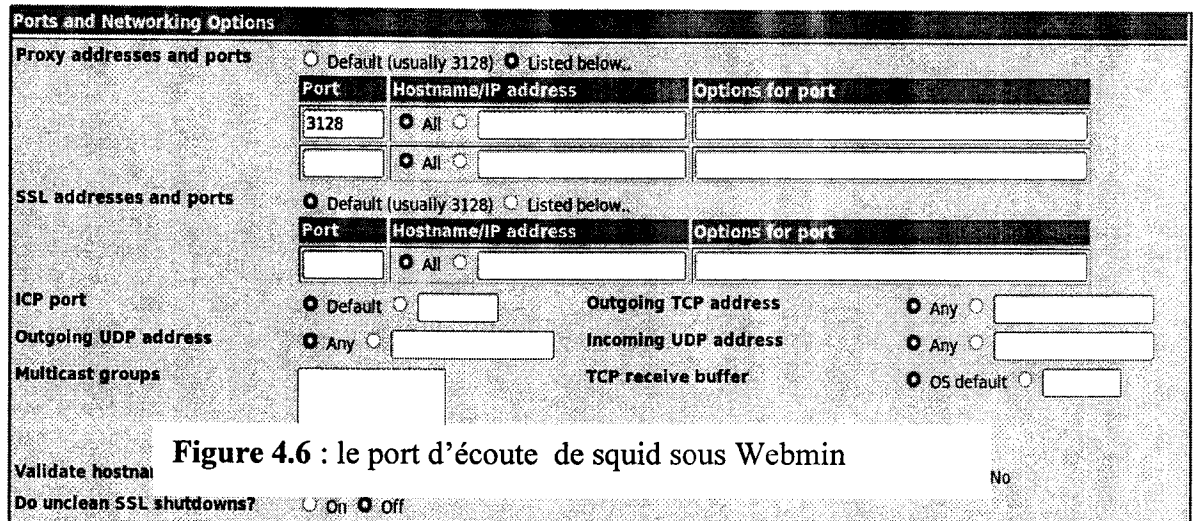


Figure 4.6 : le port d'écoute de squid sous Webmin

Par défaut le port est 3128 mais on peut le changer et mettre un numéro de port arbitraire et aussi définir une adresse IP valide pour notre interface d'écoute.

3.3. Configuration du Cache

Dans cette étape on définit la taille du cache et les étages. Dans cette partie on peut délimiter la taille des fichiers à télécharger par les utilisateurs.

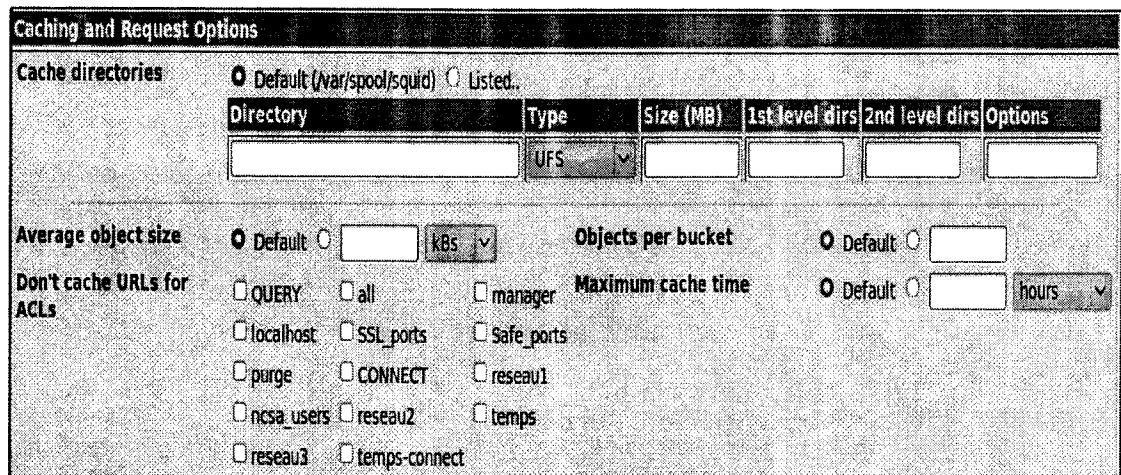


Figure 4.7 : Le cache proxy du squid sous Webmin

4. Définition des ALC

On définit les ACLs Suivants : les types de paramètres

Access control lists			Proxy restrictions	ICP restrictions	External ACL programs	Reply proxy restrictions
Name	Type	Matching..				
QUERY	URL Path Regexp	cgi-bin ?				
all	Client Address	0.0.0.0/0.0.0.0				
manager	URL Protocol	cache_object				
localhost	Client Address	127.0.0.1/255.255.255.255				
SSL_ports	URL Port	443 563 10000				
Safe_ports	URL Port	80				
Safe_ports	URL Port	21				
Safe_ports	URL Port	443 563				
Safe_ports	URL Port	70				
Safe_ports	URL Port	210				
Safe_ports	URL Port	1025-65535				
Safe_ports	URL Port	280				
Safe_ports	URL Port	488				
Safe_ports	URL Port	591				
Safe_ports	URL Port	777				
Safe_ports	URL Port	901				
purge	Request Method	PURGE				
CONNECT	Request Method	CONNECT				
reseau1	Client Address	192.168.2.0/255.255.255.0				
nicsa_users	External Auth	REQUIRED				
reseau2	Client Address	192.168.1.0/255.255.255.0				

Create new ACL Browser Regexp

Figure 4.8 : ACL de squid sous Webmin

On crée une ACL avec l’option “créer une nouvelle ACL” et on choisit ce qu’on veut interdire on a une liste déroulante qui propose plusieurs types de restrictions. On va s’intéresser aux heures de connexions des utilisateurs. Mais avant tout il est préférable de définir notre réseau Local où ces restrictions vont s’appliquer.

4.1. Les contrôles d’accès au proxy squid

Access control lists					Proxy restrictions	ICP restrictions	External ACL programs	Reply proxy restrictions
Action	ACLs							Move
<input type="checkbox"/> Deny	manager							↓
<input type="checkbox"/> Allow	purge localhost							↓↑
<input type="checkbox"/> Deny	purge							↓↑
<input type="checkbox"/> Deny	CONNECT SSL_ports							↓↑
<input type="checkbox"/> Allow	localhost							↓↑
<input type="checkbox"/> Allow	CONNECT							↓↑
<input type="checkbox"/> Deny	!nicsa_users							↓↑
<input type="checkbox"/> Allow	reseau2							↓↑
<input type="checkbox"/> Allow	reseau3							↓↑
<input type="checkbox"/> Allow	temps-connect							↓↑
<input type="checkbox"/> Allow	reseau1							↓↑
<input type="checkbox"/> Deny	all							↑

Figure 4.9 : contrôle d’accès au proxy

5. Définition du réseau local

Voilà ACL de notre réseau 1 : @IP qu'on veut filtrer : réseau de serveur web

Client Address ACL

ACL Name: reseau1

From IP: 192.168.2.0 To IP: Netmask: 255.255.255.0

Failure URL:

Store ACL values in file: Squid configuration Separate file

Figure 4.10 : ACL réseau 1

Voilà ACL de notre réseau 2 : (réseau local)

Client Address ACL

ACL Name: reseau2

From IP: 192.168.1.0 To IP: Netmask: 255.255.255.0

Failure URL:

Store ACL values in file: Squid configuration Separate file

Figure 4.11 : ACL réseau 2

6. Définition des ALCs pour les heures de connexions

On veut autoriser la connexion des clients par exemple de 14:00-17:00 tous les jours de la semaine :

Exemple de configuration avec Webmin:

Le nom de acl est : temps-connect

Date and Time ACL

ACL Name: temps-connect

Days of the Week: All Selected

Hours of the Day: All 14:00 to 17:00

Failure URL:

Store ACL values in file: Squid configuration Separate file

Figure 4.12 : ACL autorisé connexion entre 14-17

On configure de la même manière pour les deux autres ACLs et on passe à leur application.

Même on peut interdire toutes les machines du réseau 2 de faire une connexion au serveur web.

7. Application des ACLs

Une fois les ACLs créés on passe à leur application toujours dans la même fenêtre on ouvre l'onglet "Restrictions du proxy" et on ajoute nos ACLs.

Action	ACLs	Move
<input type="checkbox"/> Deny	manager	↓
<input type="checkbox"/> Allow	purge localhost	↓↑
<input type="checkbox"/> Deny	purge	↓↑
<input type="checkbox"/> Deny	CONNECT SSL_ports	↓↑
<input type="checkbox"/> Allow	localhost	↓↑
<input type="checkbox"/> Allow	CONNECT	↓↑
<input type="checkbox"/> Deny	ncsa_users	↓↑
<input type="checkbox"/> Allow	reseau2	↓↑
<input type="checkbox"/> Allow	reseau3	↓↑
<input type="checkbox"/> Allow	temps-connect	↓↑
<input type="checkbox"/> Allow	reseau1	↓↑
<input type="checkbox"/> Deny	all	↓↑

Figure 4.13 : ACL temps-connect est allow

Pour ajouter une ACL on doit savoir si on doit autoriser la règle ou l'interdire dans notre cas nous avons définie des heures ou des utilisateurs pouvaient se connecter donc on va autoriser Implicitement il y a un "deny all" qui se trouve en bas des restrictions donc ce qui veut dire que tout ce qui n'est pas autorisé est interdit.

Webmin 1.550 on serveur.mast...
 Login: root
 Module Index
 Create Proxy Restriction

Proxy Restriction
 Action: Allow Deny

Match ACLs	Don't match ACLs
all	all
manager	manager
localhost	localhost
SSL_ports	SSL_ports
Safe_ports	Safe_ports
purge	purge
CONNECT	CONNECT
reseau1	reseau1
ncsa_users	ncsa_users

Save
 Return to ACL list | Return to index

Figure 4.14 : ACL deny est allow

7.1. Définition de l'authentification du nom d'utilisateur / mot de passe

La meilleure configuration est que chacune des adresses IP peut accéder à notre proxy Squid, nous allons utiliser l'authentification par mot de passe.

Authentification client

Pour pouvoir être authentifié par Squid il faut tout d'abord créer notre fichier d'utilisateurs. Pour cela, effectuez les commandes suivantes :

```
touch /etc/squid/users
```

Pour créer le fichier en ajoutant un utilisateur :

```
htpasswd -c /etc/squid/users <nom de l'utilisateur>
```

Pour les utilisateurs supplémentaires, au-delà du premier :

```
htpasswd /etc/squid/users <nom de l'utilisateur>
```

Il faut répéter la dernière opération autant de fois que vous avez d'utilisateur. Sachez que les mots de passe sont cryptés avec la méthode *crypt*

Pour vérifier que le fichier est correcte et fonctionnera avec Squid utilisez la commande suivante :

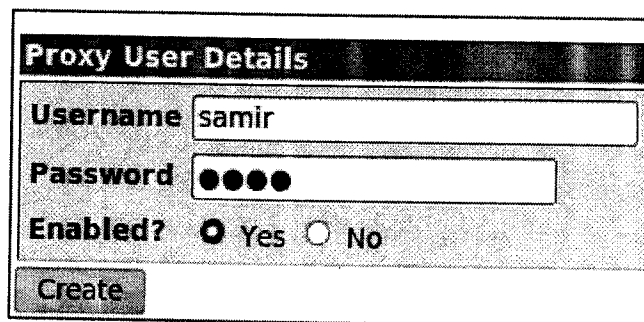
```
/usr/lib/squid/ncsa_auth /etc/squid/users
```

Rentrez votre login et votre mot de passe de la manière suivante :

```
<login> <mot de passe>
```

Si la réponse est **OK** tout est bon, si la réponse est **ERR** vérifiez votre fichier users.

Voilà la création d'un nouvel utilisateur pour la connexion au proxy squid par webmin



Proxy User Details	
Username	samir
Password	●●●●●
Enabled?	<input checked="" type="radio"/> Yes <input type="radio"/> No
<input type="button" value="Create"/>	

Figure 4.15 : utilisateur de connexion au proxy

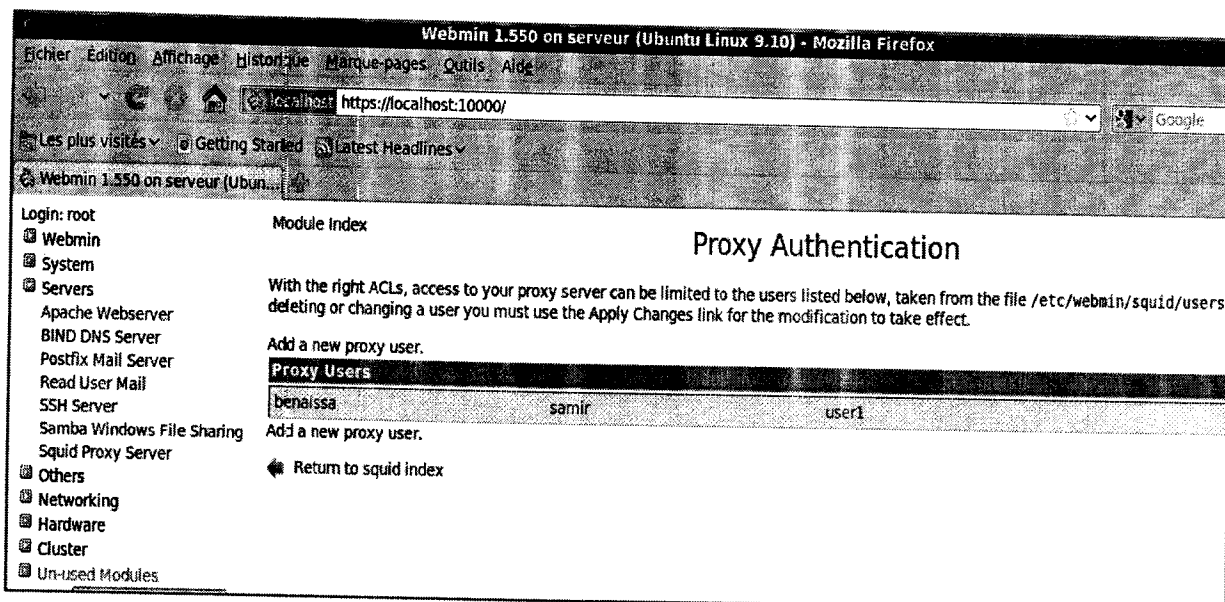


Figure 4.16: authentification par les utilisateurs de proxy

Configuration de Squid

« Éditez le fichier `/etc/squid/squid.conf` de configuration de Squid, « ¹⁷

dans la partie TAG: `auth_param`, décommenter les lignes suivantes : `auth_param` et remplacer les ci besoin par celles ci-dessous :

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/users
```

```
auth_param basic children 5
```

```
auth_param basic realm Squid proxy-caching web server
```

```
auth_param basic credentialsttl 2 hours
```

children : 5 est une valeur usuelle. Si vous avez de nombreux utilisateurs, il sera peut-être nécessaire d'augmenter ce nombre.

realm : texte qui apparaîtra dans la fenêtre de demande d'identification.

credentialsttl : durée de vie de l'identification.

Ensuite, dans la partie `acl` rajouter la ligne suivante :

```
acl Users proxy_auth REQUIRED
```

Enfin, dans la partie `http_access` rajouter la ligne suivante :

```
http_access allow GroupePC Users
```

GroupePC représente le groupe de machine que vous avez autorisé à avoir accès à votre proxy, pour plus de renseignement il faut se référer au site de [Squid](http://www.squidguard.org/config/)

Pour finir il faut relancer Squid :

```
sudo /etc/init.d/squid restart
```

¹⁷ <http://www.squidguard.org/config/>

Et voilà le résultat de système d'authentification sous linux

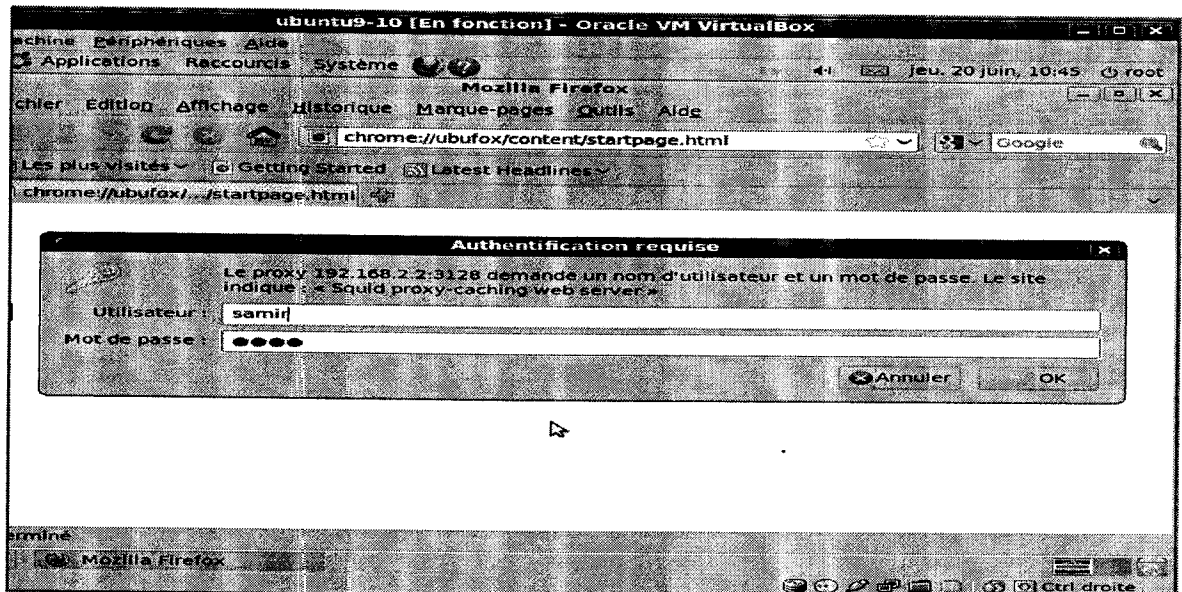


Figure 4.17 : système d'authentification d'un proxy squid sous linux

Authentification proxy sous Windows 7

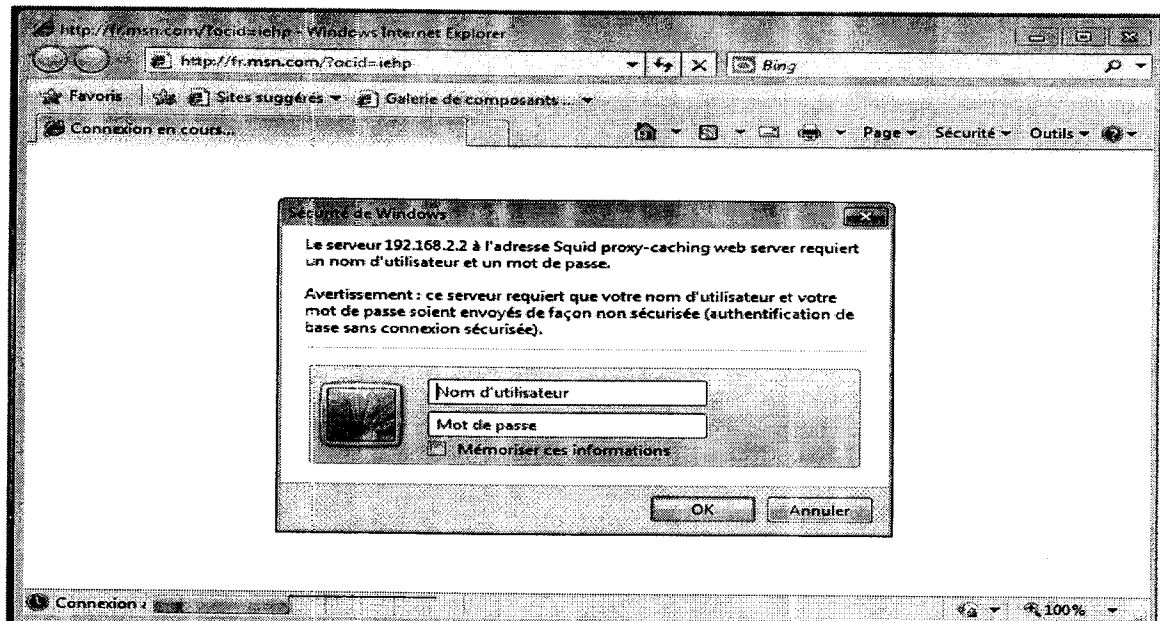


Figure 4.18 : système d'authentification d'un proxy squid sous Windows

Ouverture à l'extérieur

On accepte de fournir le service à l'extérieur, et pour cela on va insérer une règle « iptables »

```
iptables -A INPUT -p tcp --dport 3128 --sport 1024:65535 -s 192.168.2.0/24 -j  
ACCEPT
```

Ouverture du squid.

Pour ouvrir le squid filtrant à l'extérieur, il faut autoriser cet accès qui se ferme par défaut:

```
acl clients_autorises ip 192.168.2.0/24
```

On applique alors cet acl à la restriction `http_access`

```
http_access allow clients_autorises
```

```
http_access deny all
```

Ouverture du squidguard.

Une fois ceci fait, on va s'occuper des restrictions squidGuard :

```
src clients_autorises {  
    ip 192.168.2.0/24  
}
```

8. Configuration de squidguard

La configuration de squidguard est réalisée dans le fichier `squidguard.conf`. Il est en générale installé dans le répertoire `/etc/squid/squidguard.conf`

Pour expliquer le fonctionnement, nous allons présenter quelques types de configurations.

Configuration élémentaire

Par défaut, on écrit le fichier de configuration suivant qui autorise tout le monde à accéder à l'ensemble des sites :

```
logdir /usr/local/squidGuard/log  
acl {  
    default {  
        pass all  
    }  
}
```

Configuration permettant d'éliminer les domaines

La configuration suivante permet de rendre des sites Internet inaccessible en les sélectionnant dans un fichier :

```
logdir /usr/local/squidGuard/log
```

```
dbhome /usr/local/squidGuard/db
dest local {
domainlist local/domains
}
acl {
default {
pass local none
redirect http://localhost/refu.htm
}
}
```

Dans le fichier "/usr/local/squidGuard/db/local/domains", on indiquera la liste des sites refusés : par exemple : licence-pfe.dz

Configuration autorisant certains clients à utiliser Squid

```
logdir /usr/local/squidGuard/log
dbhome /usr/local/squidGuard/db
src privileged {
ip 192.168.1.0
ip 192.168.2.0-192.168.2.20
ip 10.0.1.32/27
ip 10.0.2.0/255.255.255.0
domain licence-pfe.dz
}
acl {
privileged {
pass all
}
default {
pass none
redirect http://localhost/refu.htm
}
}
```

Sélection des domaines et des url non autorisés

On interdit l'accès aux domaines cités dans le fichier games/domains et les url cités dans le fichier games/urls. Voici le fichier squidguard.conf :

```
logdir /usr/local/squidGuard/log
dbhome /usr/local/squidGuard/db
dest games {
domainlist games/domains
urllist games/urls
}
acl {
default {
pass !jeux all
redirect http://localhost/refu.htm
}
}
```

Autoriser l'accès à certains sites à certains clients

Exemple :

```
logdir /usr/local/squidGuard/log
dbhome /usr/local/squidGuard/db
src direction {
192.168.1.0/24
user foo bar
}
src administratif {
ip 192.168.2.0/24
}
dest jeux {
domainlist jeux/domains
urllist jeux/urls
}
acl {
direction {
```

```
pass all
}
administratif {
pass !jeux all
}
default {
pass none
redirect http://localhost/refu.htm
}
}
```

Cette configuration permet d'autoriser les membres de direction à se connecter où bon leur semblent et d'empêcher aux membres administratifs l'accès aux sites et url de jeux précisés dans les fichiers : /usr/local/squidguard/db/jeux/domains et /usr/local/squid/db/jeux/urls.

9. Conclusion

Un squid est un moyen très important pour filtrer les requêtes des clients de notre réseau local, et mettre des règles de restriction à l'accès à internet avec l'authentification des clients. Nous avons configuré et administré ce service par une application très performante et efficace s'appelant Webmin (gestionnaire graphique des serveurs réseau). Le squid complète les autres moyens de sécurité comme netfilter et les réseaux privés virtuel vpn dans le domaine réseau. Cette configuration est suivie par des différents tests et règles de restriction qui ont été réalisés dans un environnement virtuel décrit par l'outil virtualbox.

Les services Web sont devenus de plus en plus populaires, les utilisateurs souffrent des problèmes de surcharge des serveurs et de congestion du réseau. Le proxy est reconnu comme étant une technique efficace pour diminuer la congestion des serveurs, et de réduire le trafic réseau, et de cette façon, minimiser le temps de latence des utilisateurs.

Le proxy garde les pages consultées en mémoire afin de les redonner directement aux clients s'ils tentent d'y accéder directement.

Cette fonction permet d'augmenter considérablement la rapidité de chargement des pages les plus consultées mais aussi des images constituant les styles des sites que nous visitons le plus souvent et qui représentent souvent la charge la plus lourde sur un site.

Le serveur proxy http étant idéalement très proche du client, ce téléchargement s'effectue à vitesse normale pour l'utilisateur.

Nous sommes intéressés dans notre mémoire par la configuration et administration d'un proxy squid dans un environnement virtuel. Ce projet nous a permis de découvrir les performances et l'utilité d'un proxy dans un réseau intranet. C'est aussi un moyen très important pour filtrer les requêtes des clients de notre réseau local, et mettre des règles de restriction de l'accès à internet avec l'authentification des clients.

Comme perspectives dans notre travail, nous voulons avoir une continuité dans notre projet vers une étude des performances de serveur proxy cache coopératif.

- [1] Laurent Bloch et Christophe Wolfhugel, « Sécurité Informatique : Principes et méthode », Juin 2011.
- [2] Loïc Thomas, Squid, Intégrez un proxy à votre réseau d'entreprise, édition , Eni 2012
- [3] Kulbir Saini, Squid Proxy Server 3.1 Beginner's Guide.
- [4] Jean-François Apréa, Hyper-V (version 3) et SC Virtual Machine Manager, Technologie de virtualisation sous Windows Server 2012
- [5] Eric Maillé, Damien Bruley et René-François, Les solutions de virtualisation au sein de votre organisation (serveur et poste de travail), Eni, 2012.
- [6] David Crowder , Créer un site Web pour les nuls, First ,2011.
- [7] José Dordoigne , Réseaux informatiques, Notions fondamentales et Administration sous Windows ou Linux - Théorie et Pratique - 2 livres en 1.
- [8] Aymen Mellassine, « Systèmes de Détection d'Intrusions dans les Réseaux Ad Hoc», Rapport de projet de fin d'études, Ecole supérieur de télécommunication de Tunis, 2005.
- [9] Guy Pujolle, « Les réseaux », Eyrolles, 2008.
- [10] Pål Baltzersen, Lars-Erik Häland Stage Administrateurs Réseaux ; D2 DIANA Eric Gindre mars 2005 modif janvier 2006
- [11] Article > GNU /Linux : Serveur mandataire.
- [12] Tunisien team ; Wajih Letaief Virtualisation sous Ubuntu avec VirtualBox novembre 2008.
- [13] Michel Frenkiel et al. « Les enjeux de sécurité », Septembre 2009.
- [14] Liran Lerman, « Mise en place d'un proxy Squid avec authentification Active Directory », Thèse de doctorat, Université limôge, 2008.
- [15] Proxy Squid & SquidGuard Philippe Latu philippe.latu(at)inetdoc.net
- [16] Ferro Luca et Salman Nader, « Sécurité Réseaux », Université de Nice Sophia-Antipolis, 2006.
- [17] Abi-ayyed salima « Etudier une solution de virtualisation par conteneur sous linux », Rapport de stage de perfectionnement de personnel de CRI, Nice Sophia-Antipolis ,2011.

Références :

- Site de SquidGuard.conf : <http://www.squidguard.org>
- Site de configuration de Squidguard : <http://www.squidguard.org/config/>
- Docs intéressante sur Squid : <http://linux.crdp.ac-caen.fr/Docs/Squid/t1.html>

Figure 1.1 : machine virtuelle	5
Figure 1.2 : virtualisation complète	7
Figure 1.3 : para-virtualisation	9
Figure 1.4 : hyperviseur	10
Figure 1.5 : réseau virtuel	12
Figure 2.1 : serveur web	14
Figure 2.2 : hébergement d'un serveur web.....	15
Figure 2.3 : schéma d'un serveur web	16
Figure 2.4 : serveur ftp	19
Figure 2.5 : mode ftp actif	20
Figure 2.6 : mode ftp passif	21
Figure 3.1 : serveur proxy	24
Figure 3.2 : cache d'un serveur proxy	25
Figure 3.3 : filtrage d'un serveur proxy	25
Figure 3.4 : authentification par un serveur proxy	26
Figure 3.5 : page Access denied.....	30
Figure 4.1 : Trois machines virtuelles sous virtualbox	34
Figure 4.2 : architecture de notre réseau	35
Figure 4.3 : configuration la connexion à un serveur proxy	36
Figure 4.4 : Interface graphique de webmin	37
Figure 4.5 : les fonctions de squid sous webmin	37
Figure 4.6 : le port d'écoute de squid sous webmin	38
Figure 4.7 : Le cache proxy de squid sous webmin	38
Figure 4.8 : ACL de squid sous webmin.....	39
Figure 4.9 : contrôle d'accès au proxy.....	39
Figure 4.10 : ACL réseau 1.....	40
Figure 4.11 : ACL réseau 2	40
Figure 4.12 : ACL autorisé connexion entre 14-17.....	40
Figure 4.13 : ACL temps-connect est allow	41
Figure 4.14 : ACL deny est allow	41
Figure 4.15 : utilisateur de connexion au proxy.....	42
Figure 4.16 : authentification par les utilisateurs de proxy.....	43
Figure 4.17 : système d'authentification d'un proxy squid sous linux.....	44
Figure 4.18 : système d'authentification d'un proxy squid sous windows.....	44

CPU	<i>Central Process Unit</i>
FTP	<i>File Transport Protocol</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
LAN	<i>Local Area Network</i>
XML	<i>eXtensible Markup Language</i>
WAN	<i>Wide Area Network</i>
URL	<i>Uniform Resource Locator</i>
TCP	<i>Transmission Control Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Sécurisé</i>

Résumé

Le serveur Squid est la solution de proxy cache pour le Web la plus connue et la plus utilisée. L'intérêt majeur d'un proxy, sur un réseau local est de permettre de donner la connectivité Internet à l'ensemble des machines locales tout en les maintenant invisible de l'extérieur. Cette centralisation permet de contrôler le trafic Web et de filtrer les protocoles.

Nous sommes intéressés dans notre mémoire par la configuration et administration d'un proxy squid dans un environnement virtuel. Il donne des performances intéressantes à notre réseau intranet. C'est un moyen très important pour filtrer les requêtes des clients et mettre des règles de restriction de l'accès à internet avec un système d'authentification et de sécurité.

Mot clés : *squid, squidgard, serveur apache, serveur ftp, acl, proxy, cache proxy.*

Abstract

The Squid server solution proxy cache for the best known and most widely used Web. The major advantage of a proxy on a local network is to provide internet connectivity to all local machines while keeping them invisible from the outside. This centralization of control Web traffic and filter protocols.

We are interested in our memory configuration and administration of a squid proxy in a virtual environment. It gives interesting performances to our intranet. This is a very important way to filter client requests and implement rules restricting internet access with authentication and security.

Keyword: *squid, squidgard, serveur apache, ftp server acl, proxy, proxy cache.*