

Ministère de l'enseignement supérieur de la recherche scientifique

Université Ibn Khaldoun -Tiaret

Faculté des sciences et science de l'ingénieur

Département informatique

Mémoire de fin d'études

En vue de l'obtention du diplôme master

En informatique option réseau et télécom

Thème

Réalisation d'un pare-feu sécurise sous proxy sous linux

présenté par:

Mostefa zineb

RACHEDI HABIBA

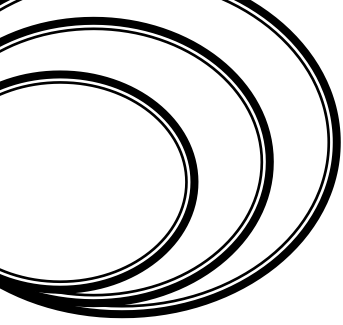
encadré par

M. BENGHANI MALIK

Année universitaire 2011-2012

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



Remerciements

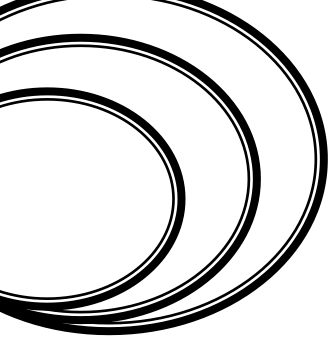
Avec l'aide d'ALLAH est achevé le présent travail, cependant nous tenons à exprimer nos sincères remerciements à certaines personnes dont les conseils et encouragements ont été précieux durant la réalisation de ce travail.

Nous n'oublions pas notre encadreur monsieur « bengahni-abdelamlik » et nous remercies aussi nos enseignants tout au long du cycle d'études au département D'informatique.

Nous adressons une pensée particulièrement affective à « Abdallah Mohamed » et nos amis, qui ont rendu agréables nos longues années d'études.

Nous remercions tout particulièrement les membres du jury, pour avoir accepté de participer au jury de notre mémoire.

Nous tenons enfin à remercier tous ceux qui ont participés de près ou de loin à l'élaboration de ce travail. Que tous ceux –ci se reconnaissent et acceptent nos humbles remerciements.



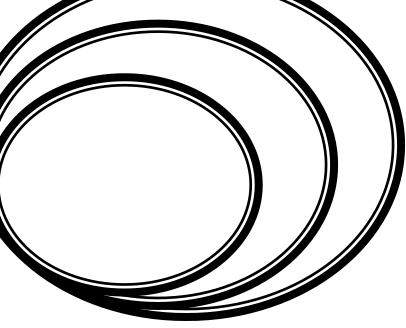
Dédicaces

On dédie ce modeste travail :

- ❖ *A mes parents.*
- ❖ *A mes frères.*
- ❖ *A toute la famille.*
- ❖ *A tous mes amis.*

Habiba, Zeineb

b



SOMMAIRE

résumé.....	1
Introduction général.....	2

CHAPITRE I « LES SYSTEMES PARE-FEU »

I.1 INTRODUCTION	4
I.2 PARE-FEU	4
I.2.1 DEFINITION D'UN PARE-FEU :	4
I.2.2 TERMINOLOGIE.....	5
I.2.3 ORIGINE DU TERME.....	5
I.2.4 POURQUOI UTILISER UN PARE-FEU?	6
I.3 FONCTIONNEMENT D'UN SYSTEME PARE-FEU.....	6
I.4 CATEGORIES DE PARE-FEU.....	6
I.4.1 PARE-FEU SANS ETAT (STATELESS FIREWALL)	6
I.4.2 PARE-FEU A ETATS (STATEFUL FIREWALL)	7
I.4.3 PARE-FEU IDENTIFIANT.....	7
I.4.4 PARE-FEU PERSONNEL.....	7
I.4.5 PARE-FEU CAPTIF.....	8
I.4.6 PARE-FEU APPLICATIF	8
I.5 PRINCIPALES CARACTERISTIQUES D'UN FIREWALL	8
I.5.1 FONCTIONS DE FILTRE ET DE CLOISONNEMENT.....	9
I.5.2 Fonctions de relais et de masque.....	10
I.6 PRINCIPE DE PROXY.....	11
I.7 DEROULEMENT DU PROXY	12
I.8 CARACTERISTIQUES DE PROXY.....	12
I.9 CRITERES DE CHOIX D'UN FIREWALL	13
CONCLUSION.....	14

CHAPITRE II «SECURITE DES RESEAUX INFORMATIQUES »

II.1 INTRODUCTION	15
II.2 DEFINITION DE BASE	15
II.3 SECURISER LES DONNEES	15
II.4 POURQUOI LES SYSTEMES SONT VULNERABLES.....	15
II.5 QU'ESSAYEZ-VOUS DE PROTEGER ?	16
II.5.1 VOS DONNEES	16
II.5.2 VOS RESSOURCES.....	16
II.5.3 VOTRE REPUTATION.....	17
II.6 CONTRE QUOI ESSAYEZ-VOUS DE COUS PROTEGER ?	17
II.6.1 TYPES D'ATTAQUES.....	17
II.6.2 TYPE D'AGRESSEURS.....	19
II.7 COMMENT POUVEZ-VOUS PROTEGER VOTRE SITE ?	20
CONCLUSION.....	21

CHAPITRE III « LES OUTILS DE CONFIGURATION DU FIREWALL »

III.1 INTRODUCTION.....	22
III.2 DEFINITION LINUX.....	22
III.2.1 Debian GNU/LINUX.....	22
III.2.3 Avantages de système linux.....	22
III.3 SQUID.....	23
III.3.1 le logiciel.....	23
III.3.2 PRINCIPE DE FONCTIONNEMENT.....	23
III.3.3 objectif.....	24
III.4 NETFILTER.....	24
III.5 ARCHITECTURE NETFILTER.....	24
III.6 FONCTIONNEMENT.....	ERREUR ! SIGNET NON DEFINI.
III.6.1 LES TABLES, CHAINES ET CIBLES.....	ERREUR ! SIGNET NON DEFINI.
III.7 PRESENTATION D'IPTABLES.....	27
III.7.1 LE SUIVI DE CONNEXION (CONNECTION TRACKERS)	28
CONCLUSION.....	29

Chapitre IV « La configuration d'un firewall »

IV.1 INTRODUCTION	ERREUR ! SIGNET NON DEFINI.
IV.2 LE BUT	ERREUR ! SIGNET NON DEFINI.
IV.3 CONFIGURATION AVEC IP STATIQUE	ERREUR ! SIGNET NON DEFINI.
IV.4 serveur DHCP	31
IV.5 INSTALLATION SQUID	ERREUR ! SIGNET NON DEFINI.
IV.5.1 CONFIGURATION MINIMALE	ERREUR ! SIGNET NON DEFINI.
IV.5.2 CREER UNE ACL REPRESENTANT LE LAN	ERREUR ! SIGNET NON DEFINI.
IV.6 CONFIGURATION DE CACHE :	ERREUR ! SIGNET NON DEFINI.
IV.7 IDENTIFIER LES UTILISATEURS	36
IV.7.1 CONSTRUIRE UN FICHIER D'UTILISATEURS	36
IV.7.2 CONFIGURER SQUID POUR RECLAMER L'IDENTIFICATION DE VOS UTILISATEURS	38
IV.8 ÉTAPES DE CONFIGURATION DU FIREWALL	39
IV.9 CONFIGURATION PROXY COTE CLIENT :	41
Conclusion	Er
<i>reur ! Signet non défini.</i>	
Annexe : Mise en ouvre de la topologie réseau	44
Conclusion générale	49
Bibliographie	50

Liste des figures

Chapitre I « Les pare-feu »

Figure I.1: Architecture d'un firewall.....	4
Figure I.2 : Fonction de filtre et de cloisonnement d'un firewall.....	9
Figure I.3 : Les différentes possibilités de filtrage d'un firewall.....	9
Figure I.4 : Fonctions de relais et de masque d'un firewall	10
Figure I.5 : Exemple de firewall proxy.....	11
Figure I.6 : Déroulement du proxy.....	12

Chapitre III « Les outils de configuration d'un firewall »

Figure III.1 : Architecture de Netfilter.....	25
Figure III.2 : connection Trackers	29

Chapitre IV « La configuration d'un firewall»

Figure IV.1: Interface de web des erreurs d'authentification d'un client.....	32
Figure IV.2 : l'accès d'un client autorisé à une page web.....	35
Figure IV.3 : demande d'authentification.....	39

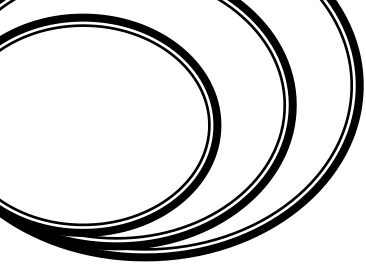
Annexe

Figure.1 : Topologie du réseau.....	44
Figure.2 : Les machines virtuelles.....	45
Figure .3 : Attachement de l'interface de connexion du client.....	46
Figure .4 : Attachement de l'interface de connexion du perfeu.....	47

Liste des tables

Chapitre III « Iptables /Netfilter »

Tab III.1 : Tables principales de Netfilter.....	26
Tab III.2 : chaînes de Netfilter.....	26
Tab III.3 : cibles de Netfilter.....	27
Tab III.4 : Option d'Iptables.....	28



Résumé

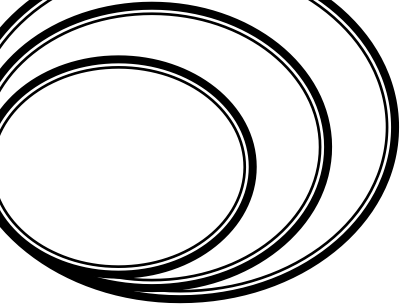
La mise en place de la sécurité au d'un réseau exige de définir une politique garantissant son efficacité optimale. Celle-ci constitue un document de référence duquel découlent les règles de filtrage et de contrôle d'accès. A ce titre la politique de sécurité permet de formaliser les droits par utilisateur ou par rôle de l'individu au sein de l'organisation.

En pratique, les droits d'accès d'un utilisateur sont implémentés par des firewalls. Les firewalls sont des entités de niveau application offrant une meilleure sécurité basée sur l'identité de l'utilisateur (représentée le plus souvent par un username {Password}, un certificat ou une signature biométrique) ainsi que d'autres paramètres tel que le temps d'accès et l'adresa IP.

Les firewalls agissent au niveau de la couche transport et s'exécutent dans l'espace noyau du système d'exploitation et ils sont donc d'un coté très performant.

L'objectif de ce travail est de réaliser un firewall fonctionnant au niveau application tout en ayant la possibilité d'identifier les flux de chaque utilisateur

Mots clés : *firewall, proxy, sécurité, squid.*



Introduction générale

Il devient impossible d'entrer dans une bibliothèque, de lire un magazine ou d'écouter la radio sans voir ou entendre quelque chose sur l'Internet. Celui-ci est devenu si populaire qu'il ne requiert quasiment plus aucune explication quand on le mentionne dans des publications généralistes, qu'il s'agisse du *Nouvel Observateur* ou de modes et travaux. Alors que les revues généralistes sont actuellement obsédées par l'Internet. Elle constitue une avancée technologique remarquable qui fournit un accès à l'information, et la possibilité d'en publier soi-même, d'une manière révolutionnaire.

Cette ouverture vers l'extérieur est indispensable et dangereuse en même temps. Ouvrir l'entreprise vers le monde signifie aussi laisser place ouverte aux étrangers pour essayer de pénétrer le réseau local de l'entreprise et y accomplir des actions douteuses, parfois gratuites, de destruction, vol d'informations confidentielles, ... Les mobiles sont nombreux et dangereux.

Il s'agit également d'un danger majeur qui donne la possibilité de polluer et de détruire l'information d'une manière non moins révolutionnaire.

❖ Problématique suivante :

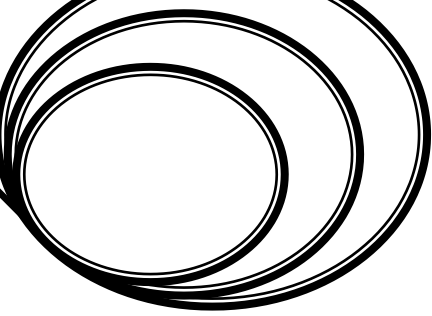
Explore le problème de sécurité Internet et se consacre aux firewalls en tant qu'élément d'une stratégie efficace destinée à résoudre ce problème.

❖ L'objectif de notre travail :

Est une contribution en vue d'améliorer la protection c'est-à-dire comment on va faire pour protéger un réseau informatique contre les attaques à partir d'un pare-feu.

❖ Ce mémoire est composé en quatre chapitres :

✓ Le premier chapitre « les systèmes pare-feu » a pour objet de: Mettre en évidence les besoins de sécurité liés au contrôle d'accès Présenter et analyser les fonctions et les caractéristiques d'un système pare-feu (firewall).



- ✓ Le deuxième chapitre « sécurité des systèmes d'informations » est basé sur:
La sécurité et les problèmes posés par une sécurité des informations lors des échanges au travers de réseaux publics ou privés, et les fonctions de sécurité sont aussi traités par ce chapitre.

- ✓ Le troisième chapitre « les outils de configuration du firewall » consiste à présenter le firewall Netfilter sous linux (distribution Debian), son architecture et son fonctionnement ainsi que la présentation de Squid.

- ✓ Le quatrième chapitre « la configuration du firewall » décrit les différentes étapes utilisées pour la configuration du firewall sous linux.

Nous terminons ce mémoire par une conclusion générale où on retrace les grandes lignes de ce travail et où on met l'accent sur les principaux résultats obtenus.

I.1 Introduction

A partir du moment où une machine d'une organisation se connecte à internet celle-ci devient vulnérable à un certain nombre d'attaques venant de l'extérieur. L'entreprise doit donc définir une politique globale de sécurité et la mettre en place.

En effet, il est nécessaire de se protéger de ces attaques pour parer à celles-ci une architecture de réseau sécurisée est nécessaire. L'organisation doit comporter un composant essentiel qui est le firewall. Cet outil a pour but de sécuriser au maximum le réseau local de l'entreprise de détecter les tentatives d'intrusion et d'y parer au mieux possible. Cela permet de rendre le réseau ouvert sur internet beaucoup plus sûr.

I.2 Pare-feu

I.2.1 Définition d'un pare-feu :

Un pare-feu (firewall en anglais), c'est un dispositif informatique qui filtre les flux d'information entre un réseau interne à l'organisme et un réseau externe afin de maîtriser les accès vers l'extérieur, en analysant les informations contenues dans les couches 2, 3, 4 et 7 du modèle OSI (Open Systems Interconnexion). Il s'agit donc d'une machine comportant au minimum deux interfaces réseau [6] [7]:

- une interface pour le réseau à protéger (réseau interne).
- une interface pour le réseau externe (internet).

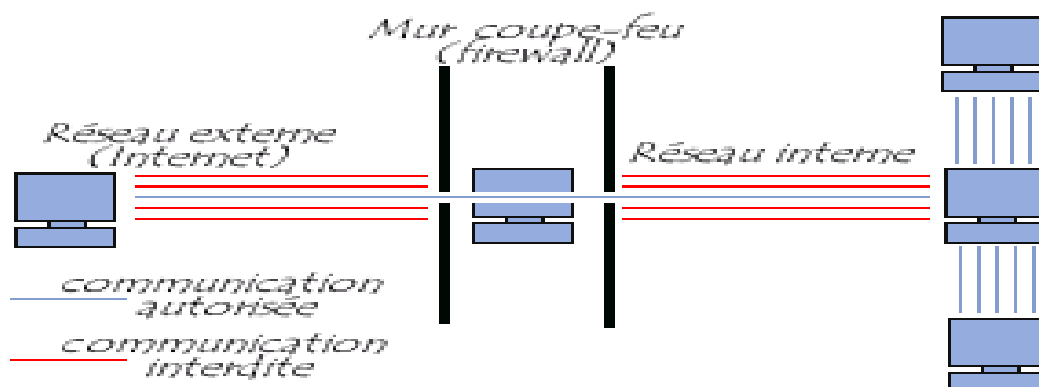


Figure I.1: Architecture d'un firewall

1.2.2 Terminologie

Un pare-feu est parfois appelé coupe-feu, garde-barrière, barrière de sécurité, ou encore firewall. Un pare-feu est aussi appelé *packet filter* [6].

1.2.3 Origine du terme

Le terme peut avoir plusieurs origines : le « *pare-feu* » ou « *coupe-feu* » est au théâtre un mécanisme qui permet, une fois déclenché, d'éviter au feu de se propager de la salle vers la scène.

Le mot fait aussi référence aux allées pare-feux qui en forêt sont destinées à bloquer les incendies de forêt, ou dans le domaine de l'architecture aux portes coupe-feux. L'usage du terme « *pare-feu* » en informatique est donc métaphorique : une porte empêchant les flammes de l'Internet de rentrer chez soi et/ou de « *contaminer* » un réseau informatique [6].

1.2.4 Pourquoi utiliser un pare-feu?

Les pare-feu sont utilisés principalement dans 4 buts [1] [6] :

➤ ***maintenir des personnes dehors :***

En effet ; pour se protéger des malveillances (externes), les firewalls permettent d'écarter divers intrus comme :

- les vandales, ceux qui veulent embêter pour embêter (saturation de liaisons, saturation de CPU, corruption de données, mascarade d'identité...).
- les espions (problème de confidentialité de l'information).

➤ ***maintenir des personnes à l'intérieur :***

Les pare-feux ont également pour objectif d'éviter la fuite d'information, non contrôlée vers l'extérieur.

➤ ***contrôler les flux :***

Tous les flux du trafic entre le réseau interne et externe doivent être surveillés .cela permet par exemple d'avoir une vue de la consommation internet différents utilisateurs internes et de bloquer l'accès à certains sites contenant des informations illégales .les garde-barrières effectuant un filtrage applicatif peuvent effectuer des vérifications sur les e-mails reçus. Enfin, un firewall permet un audit de façon (centrale) du trafic pour aider à prévoir l'évolution du réseau.

➔ **faciliter l'administration du réseau :**

Sans firewall chaque machine du réseau est potentiellement exposée aux attaques d'autres machines d'internet. Les firewalls simplifient la gestion de la sécurité et donc l'administration du réseau car ils centralisent les attaques potentielles au niveau du firewall plutôt que sur le réseau tout entier.

1.3 Fonctionnement d'un système pare-feu

Un système pare-feu contient un ensemble de règles prédéfinies permettant [1] [6]:

- D'autoriser la connexion (*allow*) ;
- De bloquer la connexion (*deny*) ;
- De rejeter la demande de connexion sans avertir l'émetteur (*drop*).

L'ensemble de ces règles permet de mettre en œuvre une méthode de filtrage dépendant de la **politique de sécurité** adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- soit d'autoriser uniquement les communications ayant été explicitement autorisées.
- soit d'empêcher les échanges qui ont été explicitement interdits.

Le choix de l'une ou l'autre de ces méthodes dépend de la politique de sécurité adoptée par l'entité désirant mettre en œuvre un filtrage des communications. La première méthode est sans nul doute la plus sûre, mais elle impose toutefois une définition précise et contraignante des besoins en termes de communication.

1.4 Catégories de pare-feu

Les pare-feu sont un des plus vieux équipements de sécurité et, en tant que tels, ils ont été soumis à de nombreuses évolutions. Suivant la génération du pare-feu ou son rôle précis, on peut les classer en différentes catégories.

1.4.1 Pare-feu sans état (stateless firewall)

C'est le plus vieux dispositif de filtrage réseau, introduit sur les routeurs. Ils font un contrôle de chaque paquet indépendamment des autres en se basant sur les règles prédéfinies par l'administrateur (généralement appelées ACL, *Access Control List*).

Ces firewalls interviennent sur les couches réseau et transport. Les règles de filtrages s'appliquent alors par rapport à une d'adresses IP sources ou destination, mais aussi par rapport à un port source ou destination [2] [7].

1.4.2 Pare-feu à états (stateful firewall)

Certains protocoles dits « à états » comme TCP introduisent une notion de connexion. Les pare-feu à états vérifient la conformité des paquets à une connexion en cours. C'est-à-dire qu'ils vérifient que chaque paquet d'une connexion est bien la suite du précédent paquet et la réponse à un paquet dans l'autre sens. Ils savent aussi filtrer intelligemment les paquets ICMP qui servent à la signalisation des flux IP [1].

Enfin, si les ACL autorisent un paquet UDP caractérisé par un quadruplet (Ip_src, port_src, Ip_dst, port_dst) à passer, un tel pare-feu autorisera la réponse caractérisée par un quadruplet inversé, sans avoir à écrire une ACL inverse. Ceci est fondamental pour le bon fonctionnement de tous les protocoles fondés sur l'UDP, comme DNS par exemple. Ce mécanisme apporte en fiabilité puisqu'il est plus sélectif quant à la nature du trafic autorisé. Cependant dans le cas d'UDP, cette caractéristique peut être utilisée pour établir des connexions directes (P2P) entre deux machines (comme le fait Skype par exemple) [9].

1.4.3 Pare-feu identifiant

Un pare-feu identifiant réalise l'identification des connexions passant à travers le filtre IP. L'administrateur peut ainsi définir les règles de filtrage par utilisateur et non plus par adresse IP ou adresse MAC, et suivre l'activité réseau par utilisateur. Plusieurs méthodes différentes existent qui reposent sur des associations entre IP et utilisateurs réalisées par des moyens variés. On peut par exemple citer authpf (sous OpenBSD) qui utilise ssh pour faire l'association. Une autre méthode est l'identification connexion par connexion, réalisée par exemple par la suite NuFW, qui permet d'identifier également sur des machines multiutilisateurs [1][2].

1.4.4 Pare-feu personnel

Les pare-feu personnels, généralement installés sur une machine de travail, agissent comme un pare-feu à états. Bien souvent, ils vérifient aussi quel programme est à l'origine des données. Le but est de lutter contre les virus informatiques et les logiciels espions [2].

1.4.5 Portail captif

Les portails captifs sont des pare-feu dont le but est d'intercepter les usagers d'un réseau de consultation afin de leur présenter une page spéciale (avertissement, charte, demande d'authentification, etc.) avant de les laisser accéder à Internet. Ils sont utilisés pour assurer la traçabilité des connexions et/ou limiter l'utilisation abusive des moyens d'accès. On les déploie essentiellement dans le cadre de réseaux de consultation Internet mutualisés filaires ou Wi-Fi [6].

1.4.6 Pare-feu applicatif

Le filtrage applicatif permet de filtrer les communications application par application. Le filtrage applicatif opère donc au niveau 7 (couche application) du modèle OSI, contrairement au filtrage de paquets simple (niveau 4). Le filtrage applicatif suppose donc une bonne connaissance des applications présentes sur le réseau, et notamment de la manière dont elle structure les données échangées (ports, etc.) [7].

Un firewall effectuant un filtrage applicatif est appelé généralement « passerelle applicative » (ou « proxy »), car il sert de relais entre deux réseaux en s'interposant et en effectuant une validation fine du contenu des paquets échangés. Le proxy représente donc un intermédiaire entre les machines du réseau interne et le réseau externe, subissant les attaques à leur place. De plus, le filtrage applicatif permet la destruction des en-têtes précédant le message applicatif, ce qui permet de fournir un niveau de sécurité supplémentaire.

Il s'agit d'un dispositif performant, assurant une bonne protection du réseau, pour peu qu'il soit correctement administré. En contrepartie, une analyse fine des données applicatives requiert une grande puissance de calcul et se traduit donc souvent par un ralentissement des communications, chaque paquet devant être finement analysé [7].

1.5 Principales caractéristiques d'un firewall

L'objet du réseau est d'offrir un maximum de connectivité et d'accès aux ressources. L'objet de la sécurité est de limiter ces accès. Ces deux objectifs concurrents et contradictoires se trouvent être ceux d'un firewall. Pour cela, et afin de satisfaire les objectifs de sécurité attendus du firewall, ce dernier implémente trois fonctions basiques : le filtrage, le masquage et le relais [3].

1.5.1 Fonctions de filtre et de cloisonnement

Dans une architecture de réseau, un firewall en renforce la sécurité en contrôlant les flux de données qui le traversent (en entrée et en sortie). Un firewall est un système qui permet de filtrer les communications qui lui parviennent, de les analyser et de les autoriser si elles remplissent certaines conditions, de les rejeter sinon (figure I.2),

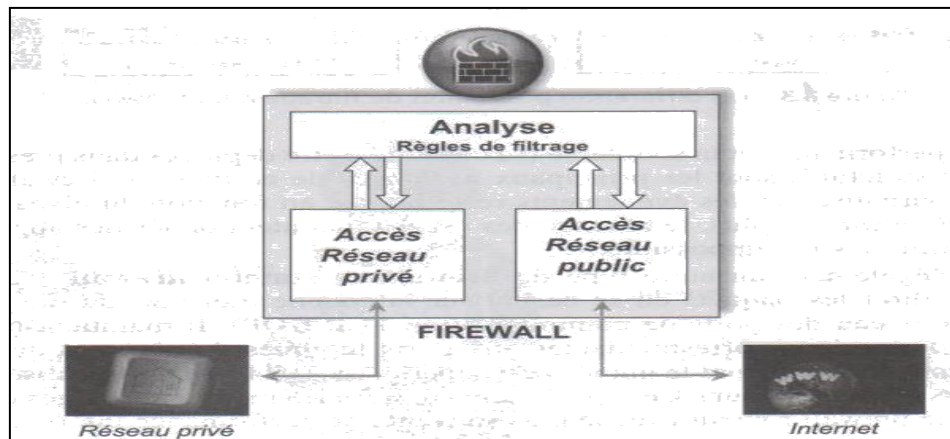


Figure I.2 : Fonction de filtre et de cloisonnement d'un firewall

Selon la nature de l'analyse et des traitements effectués par un firewall, différents types de firewalls existent. Ils se distinguent le plus souvent en fonction du niveau de filtrage des données auquel ils opèrent : niveau 3 (IP), niveau 4 (TCP, UDP) ou niveau 7 (FTP, HTTP, etc.) du modèle OSI (figure I.3) [3] [2].

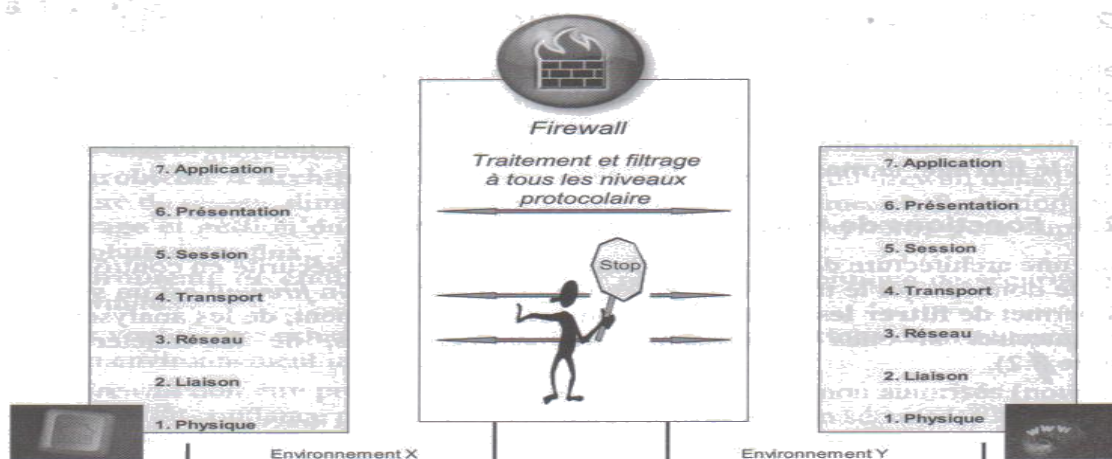


Figure I.3 : Les différentes possibilités de filtrage d'un firewall.

La performance et la souplesse (sa capacité à être déployés dans presque tout type d'infrastructure) sont les principaux avantages de ce type de firewall. Néanmoins, la journalisation des événements reste limitée en fonction du niveau auquel s'opère le firewall. En plus, le blocage des usages malintentionnés des applications est assez difficile voire impossible.

En s'interfaçant entre les systèmes du réseau d'une organisation et Internet, un firewall permet de cloisonner le réseau et éventuellement de le masquer aux utilisateurs d'Internet. Cloisonner un réseau revient à le concevoir de telle manière que l'on puisse en fonction d'impératifs de sécurité, séparer des systèmes et des environnements afin de mieux les contrôler. Le principe de cloisonnement repose sur la segmentation du système d'information en composants de Sécurité homogènes (domaines de confiance mutuelle) [6] [7].

1.5.2 Fonctions de relais et de masque

Un firewall applicatif encore dénommé proxy (serveur proxy, firewall proxy) joue un rôle de relais applicatif. Il établit en lieu et place de l'utilisateur le service invoqué par celui-ci (figure I.4),

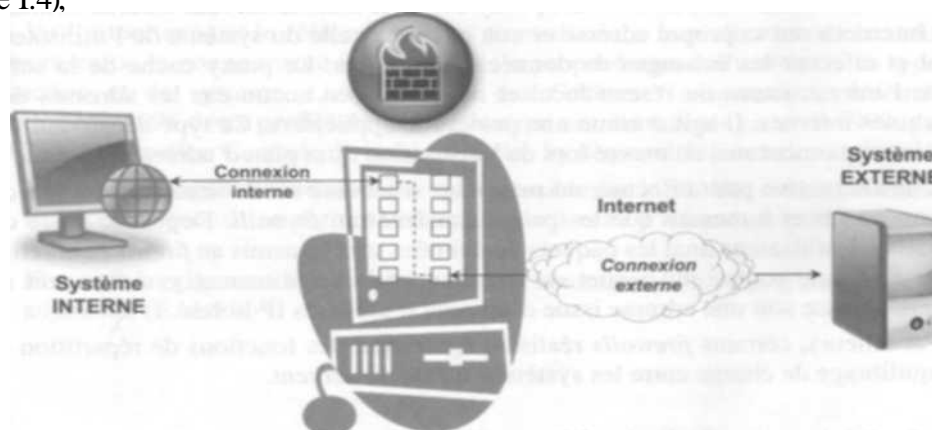


Figure I.4 : Fonctions de relais et de masque d'un firewall

L'objectif d'un système qualifié de proxy est de réaliser un masquage d'adresse car relais applicatif, et de rendre transparent l'environnement interne de l'organisation. Il est censé constituer un point de passage obligé pour toutes les applications qui nécessitent un accès Internet. Cela ne suppose qu'une application «relais» soit installée sur le poste de travail de l'utilisateur et sur le firewall (figure I.5) [3].

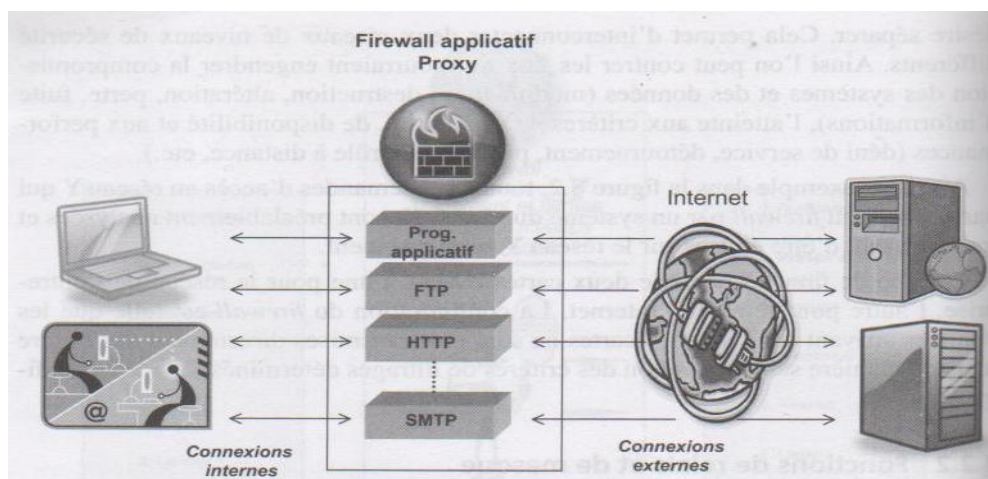


Figure I.5 : Exemple de firewall proxy

Ainsi, à chaque demande de connexion Internet, le fait de lancer un navigateur active également ce programme relais qui demandera au proxy, en son lieu et place, de réaliser la connexion externe. Le proxy contacte alors le serveur externe sollicité sur Internet avec sa propre adresse et non pas avec celle du système de l'utilisateur final et effectue les échanges de données nécessaires. Le proxy cache de la sorte toute l'infrastructure du réseau local et ne dévoile en aucun cas les adresses des machines internes. Il agit comme une passerelle applicative. Ce type de serveur est systématiquement mis en œuvre lors de l'utilisation d'un plan d'adressage privé [3].

1.6 Principe de proxy

Avec un proxy, l'arrivée d'un flux http est l'occasion de présenter une page d'authentification à l'utilisateur. Ainsi chaque utilisateur est soumis à une identification préalable avant toute initialisation de connexion par : User_Name & Password {le terme « connexion » est à prendre au sens du Conntrack de Netfilter} une fois l'identité de l'utilisateur est établie auprès du serveur, ses droits sont récupérés par Netfilter de linux. Si cette première requête est acceptée alors un temps de libre accès au réseau est accordé durant cette période, l'utilisateur est affranchi de toute identification [7].

I.7 Déroulement du proxy [7]

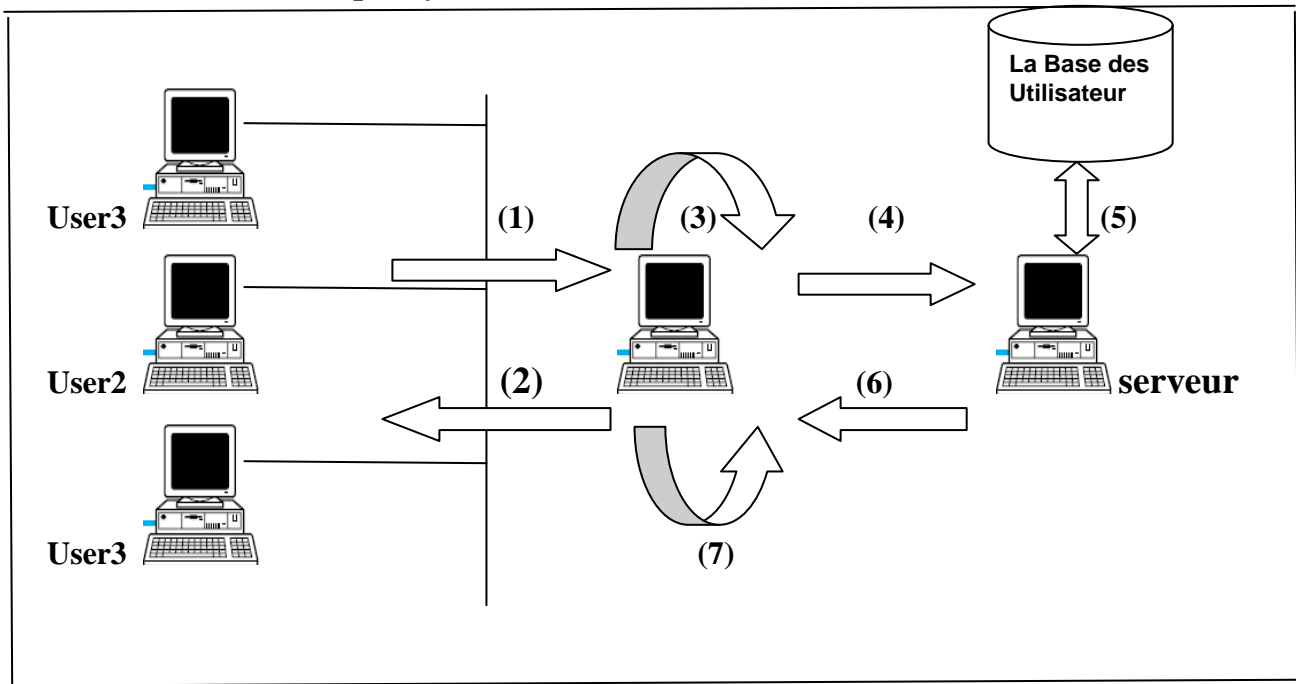


Figure I.6 : Déroulement du proxy

- (1) Arrivée d'une requête d'un utilisateur.
- (2) Le proxy affiche une page d'authentification.
- (3) Le proxy récupère le nom d'utilisateur et le mot de passe.
- (4) Le proxy construit une requête d'authentification (y compris User_Name& Password) et l'envoie au serveur.
- (5) Le serveur consulte sa base de données pour vérifier l'existence de l'utilisateur et récupère les droits qui lui sont attribués.
- (6) Le serveur envoie la réponse au proxy.
- (7) Le proxy récupère les règles à l'utilisateur pour les faire passer au Netfilter.

I.8 caractéristiques de proxy

Le firewall proxy possède les caractéristiques suivantes [7] [11] :

- Capable d'utiliser les protocoles FTP, HTTP, HTTPS.
- Il permet de mettre en cache les pages les plus fréquemment utilisées et d'accélérer de ce fait le temps de chargement desdites pages.

- Possibilité de la notion de multiutilisateur.
- Et d'améliorer la rapidité d'accès au web grâce à ses fonctions de cache.
- Définition des droits d'accès par utilisateur.
- Le contrôle et le filtrage de l'accès à la toile, en se servant des URL et éventuellement, des noms d'utilisateurs.

1.9 Critères de choix d'un firewall

Les façons de configurer un firewall et de gérer sont tout aussi importantes que les capacités intrinsèques qu'il possède. Toutefois, lorsque le choix s'impose, on prendra en considération entre autres, les critères suivants [1] [3]:

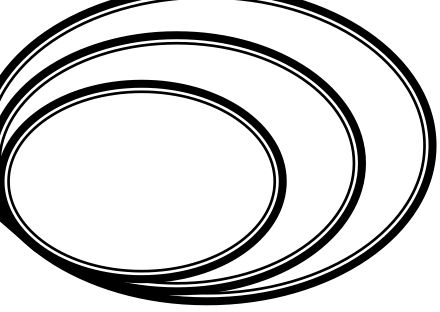
- ❖ la nature, le nombre des applications appréhendées (FTP, messagerie, HTTP, Real Audio, vidéoconférence, etc.).
- ❖ le type de filtres, le niveau de filtrage (niveau applicatif, niveau TCP, niveau IP, possibilité de combiner ces niveaux).
- ❖ les facilités d'enregistrement des actions à des fins d'audit, login. Complet des paramètres de connexion, l'existence d'outils d'analyse, d'audit actif et de détection d'activités suspectes.
- ❖ les outils et facilités d'administration (interface graphique GUI. Graphie User Interface) ou lignes de commandes, administration distante après authentification du gestionnaire, etc.).
- ❖ la simplicité du système, proxy facile à comprendre et à vérifier (facilité de configuration, etc.).
- ❖ la capacité à supporter un tunnel chiffré permettant de réaliser si nécessaire, un réseau privé virtuel (VPN, Virtual Privat Network).
- ❖ la disponibilité d'outils de surveillance, d'alarmes, d'audit actif.
- ❖ la possibilité d'effectuer de l'équilibrage de charge.
- ❖ l'existence dans l'organisation de compétences en matière d'administration du système d'exploitation du firewall.
- ❖ les intrusions dans un firewall avec pour conséquences la modification de sa configuration, des accès, l'effacement ou la modification des traces de journalisation ou encore l'infection virale.

Conclusion

On a présenté dans ce chapitre une étude les différentes Catégories de pare-feu existantes utilisées pour appliquer une politique de sécurité définie par les organisations.

On conclut, afin de garantir un niveau de protection maximal, il est nécessaire d'administrer le pare-feu et notamment de surveiller son journal d'activité afin d'être en mesure de détecter les tentatives d'intrusion et les anomalies.

La mise en place d'un firewall doit donc se faire en accord avec une véritable politique de sécurité. Dans le chapitre suivant nous allons étudier l'intérêt de la protection du firewall.



Chapitre 1 :

Les systèmes pare-feu

II.1 Introduction

La sécurité des réseaux informatiques est un sujet essentiel pour favoriser le développement des échanges dans tous les domaines. Un seul mot " sécurité " recouvre des aspects très différents à la fois techniques, organisationnels et juridiques. Latitude des utilisateurs vis à vis des problèmes de sécurité est souvent irrationnelle ce qui ne contribue pas à simplifier le débat.

D'un point de vue technique, la sécurité recouvre à la fois l'accès aux informations sur les postes de travail, sur les serveurs ainsi que le réseau de transport des données. Dans ce chapitre, nous nous concentrerons sur les problèmes posés par la sécurité des informations lors des échanges au travers de réseaux publics ou privés. Internet, le réseau des réseaux, est un outil qui permet à tous les ordinateurs quel que soit leur type de communiquer entre eux.

II.2 Définition de base

La sécurité informatique c'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles [4].

II.3 Sécuriser les données, c'est garantir [5]:

- ✓ **L'authentification** réciproque des correspondants pour être sûr de son interlocuteur
- ✓ **L'intégrité** des données transmises pour être sûr qu'elles n'ont pas été modifiées accidentellement ou intentionnellement.
- ✓ **La confidentialité** pour éviter que les données soient lues par des systèmes ou des personnes non autorisées
- ✓ **Le non répudiation** pour éviter la contestation par l'émetteur de l'envoi de données

II.4 Pourquoi les systèmes sont vulnérables [7] ?

- ✓ la sécurité est cher et difficile, les organisations n'ont pas de budget pour ça.
- ✓ les organisations acceptent de courir le risque, la sécurité n'est pas une priorité.
- ✓ la politique de sécurité est très complexe et basée sur jugements humains.
- ✓ Les systèmes de sécurité sont faits, gérés et configurés par des hommes (erreur humaine très possible).
- ✓ Il n'existe pas d'infrastructure pour les clés et autres éléments de cryptographie.
- ✓ De nouvelles technologies (et donc vulnérabilités) émergent en permanence.

II.5 Qu'essayez-vous de protéger ?

Un firewall est d'abord un système protecteur. Si vous faut d'abord vous préoccuper de ce que vous essayez de protéger. Quand vous connectez à l'Internet, vous risquez trois choses [4]:

- Vos données : les informations que vous gardez sur vos ordinateurs
- Vos ressources : les ordinateurs eux-mêmes
- Votre réputation

II.5.1 Vos données

Vos données possèdent trois caractéristiques distinctes qui justifient leur protection :

- **Le secret** : vous ne désirez probablement pas que d'autres personnes les connaissent.
- **L'intégrité** : vous ne désirez probablement pas que quelqu'un d'autre les modifie.
- **La disponibilité** : vous désirez presque certainement les utiliser vous-même.

Les gens tendent à privilégier le secret, et il est vrai qu'il s'agit généralement d'un sujet sérieux. De nombreuses organisations gardent leurs secrets les plus importants la conception de leurs produits, leurs données financières, les notes de leurs étudiants sur leurs ordinateurs. Dans certains cas, il est relativement facile de séparer les machines qui contiennent ce genre de données très secrètes des machines connectées à l'Internet.

II.5.2 Vos ressources

Même si vous avez des données qui n'ont pas d'importance pour vous, et même si vous aimez réinstaller votre système d'exploitation toutes les semaines parce que cela fait prendre de l'exercice à vos disques, si d'autres personnes utilisent vos ordinateurs, vous aimeriez probablement bénéficier de cet usage d'une façon ou d'une autre. La plupart des gens veulent utiliser leur ordinateur, ou veulent faire payer les autres pour leur usage. Même ceux qui donnent du temps du calcul et de l'espace disque en attendant généralement une certaine publicité et des remerciements ; ils n'obtiendront rien de tout cela de part d'intrus.

Vos ressources informatiques vous coutent du temps et de l'argent, et vous avez le droit de déterminer par vous-même la façon de les utiliser.

II.5.3 Votre réputation

Un intrus apparaît sur l'Internet avec votre identité. Tout se qu'il fait semble provenir de vous. Quelles en sont les conséquences ?

La plupart du temps, d'autres sites « ou des organismes de sécurité nationale » commencent à vous appeler pour vous demander pourquoi vous essayez de pénétrer dans leurs systèmes.

Quelquefois, de tels imposteurs vous coutent beaucoup plus que du temps perdu. Un intrus qui vous déteste activement, ou qui prend simplement plaisir à rendre la vie difficile à des étrangers, peut envoyer des messages électroniques ou poster des articles de news qui paraissent venir de vous. Les gens qui choisissent ceci recherchent l'animosité maximale plus que la crédibilité, mais même si seules quelques personnes croient ces messages, le nettoyage peut être long et humiliant.

II.6 Contre quoi essayez-vous de vous protéger ?

De quoi faut-il se prémunir ? Quels types d'attaques devrez-vous affronter sur l'Internet, et quels types d'attaquants sont susceptibles de les lancer ? Nous traiterons de ces sujets dans les sections suivantes, sans entrer dans des considérations techniques:

II.6.1 Types d'attaques

Il existe de nombreux types d'attaques systèmes, et nombreuses manières de les classer. Dans cette section, nous séparerons les attaques en trois grandes catégories : intrusion, refus de service et vol d'informations [4] [5] :

➤ *Intrusion*

Les attaques les plus courantes sur votre système sont les intrusions ; avec les intrusions, les gens sont réellement capables d'utiliser vos ordinateurs. La plupart des intrus veulent utiliser vos ordinateurs comme s'ils en étaient les utilisateurs légitimes.

Ces agresseurs ont à leur disposition des dizaines de manières possible d'en obtenir l'accès. Cela va de « ingénierie sociale »

➤ ***Refus de service***

Une attaque par refus de service a pour seul et unique but de vous empêcher de vous servir de vos propres ordinateurs.

Bien que certains cas de sabotage électronique impliquent la destruction ou l'invalidation du matériel ou des données, il s'agit le plus souvent de cas ressemblant à l'affaire. Un intrus inonde un système ou un réseau de tant de messages, de processus ou de requête qu'aucun travail ne peut y être effectué. Le système passe tout son temps à répondre aux requêtes et aux messages, et ne peut en satisfaire aucun.

Le risque de refus de service est la plupart du temps inévitable. A partir du moment où on accepte des communications provenant du monde extérieur « courrier électronique, appels téléphoniques ou colis postaux » on accepte la possibilité d'être submergé.

Heureusement, les attaques volontaires par refus de service ne sont pas si fréquentes que cela. Elles sont si triviales qu'elles sont méprisées par de nombreux agresseurs ; il est assez facile de remonter à la source, ce qui le rend risquées ; et elles ne fournissent ni informations, ni possibilité pour l'agresseur d'utiliser le système cible. Les attaques par refus de service délibérées sont l'œuvre de personne qui en veut à votre site en particulier, ce qui reste quand même rare.

➤ ***Vol d'informations***

Certains types d'attaque permettent à un agresseur d'obtenir des données sans en passer par l'utilisation directe de vos ordinateurs. Ces attaques exploitent en général des services Internet qui sont censés donner des informations, les conduisant à en fournir plus que ce qui est prévu, ou à les fournir à des gens non autorisés. De nombreux services Internet sont conçus pour être utilisés sur des réseaux locaux, et ne possèdent pas le type ou le degré de sécurité qui leur permettrait d'être utilisés de façon sûre sur l'Internet.

Le vol d'informations n'a même pas besoin d'être actif ou particulièrement technique. Les gens qui veulent connaître des informations personnelles pourraient se contenter de vous appeler et de vous les demander : C'est un vol d'informations. Ou bien ils peuvent espionner votre ligne téléphonique : C'est un vol d'informations passif.

II.6.2 Type d'agresseurs

Cette section décrit très brièvement les types d'agresseurs que l'on retrouve sur l'Internet. Il existe de nombreuses manières de les classer ; nous ne pouvons pas vraiment catégoriser de façon exhaustive toutes les variantes d'agresseurs que nous avons vus au cours du temps, et ce type de résumé présente nécessairement une vision stéréotypée. Il reste néanmoins utile de distinguer les principales catégories d'agresseurs [4] :

➤ *Les plaisantins*

Les plaisantins s'ennuient et cherchent à s'amuser. Ils s'introduisent chez vous parce qu'ils pensent que vous disposez d'informations intéressantes, ou parce qu'il serait amusant d'utiliser vos ordinateurs, les plaisantins sont particulièrement attirés par les sites connus et les ordinateurs inhabituels

➤ *Les vandales*

Les vandales viennent pour détruire, que ce soit pour l'excitation de la destruction ou parce qu'ils ne vous aiment pas. Quand l'un d'entre eux arrive, vous êtes immédiatement au courant.

Les vandales sont rares. Ils ne sont pas aimés, pas même des gens de l'underground qui n'ont rien contre les instruisons informatiques en générale.

Il est malheureusement impossible d'arrêter un vandale déterminé ; quelqu'un qui vous en veut vraiment vous aura à un moment au à un autre. Certaines attaques attirent les vandales mais pas d'autres types d'agresseurs ; quand les plaisantins sont dans votre système, ils ne sont intéressés que par le fait que vos ordinateurs sont en marche et disponibles depuis l'Internet.

➤ *Les compétiteurs*

De nombreux intrus sont intéressés par une version au gout du jour d'une tradition ancienne. Ils cherchent à se vanter du nombre et du type de système qu'ils ont craqué.

Les compétiteurs peuvent préférer des sites d'intérêt spécifique. Ils s'attaqueront cependant à tout ce qu'ils trouveront ; ils cherchent autant la quantité que la qualité. Ils essayeront

probablement de se donner le moyen de revenir ultérieurement. Et, si possible, ils utiliseront votre machine comme plate-forme pour attaquer d'autres sites.

➤ *Les espions (industriels et autres)*

La plupart des gens qui s'introduisent dans les ordinateurs le font pour les mêmes raisons que certains escaladent une montagne : parce qu'elle est là. Même si ces gens ne sont pas des voleurs nés, ils prennent généralement des informations qu'ils pourront directement convertir en argent ou en accès ultérieurs (carte de crédit, téléphone, ou données d'accès réseau). S'ils trouvent des secrets qu'ils pensent pouvoir négocier, ils peuvent essayer, mais ce n'est pas leur activité première.

II.7 Comment pouvez-vous protéger votre site ?

Qu'elle approche pouvez-vous essayer vous protéger contre le genre d'attaque que nous avons abordées dans cette partie ? On peut choisir entre de nombreux modèles de sécurité, de l'absence pure et simple à la sécurité réseau, en passant par ce que l'on appelle « sécurité par l'obscurité » et la sécurisation des serveurs [4] :

➤ *Absence de Sécurité*

L'approche la plus simple consiste à ne faire aucun effort en matière de sécurité, et de ne faire tourner que ce que le vendeur fournit par défaut.

Un autre modèle de sécurité possible est celui qui est communément appelé « sécurité par l'obscurité ». Dans ce modèle, on suppose qu'un système est sûr parce que (a priori) personne ne le connaît, qu'il s'agisse de son existence, de son contenu ou de quoi que ce soit d'autre. Cette approche fonctionne rarement longtemps ; il existe beaucoup trop de manière de trouver une cible attrayante.

➤ *Sécurité par L'hôte*

Le modèle le plus courant de sécurité informatique consiste à sécuriser les serveurs.

Dans ce modèle, vous renforcez séparément la sécurité de chaque machine hôte. ce n'est pas tant qu'il ne fonctionne pas sur des machines individuelles, qu'il ne correspond pas à l'échelle d'un vaste réseau.

➤ *Sécurité par réseau*

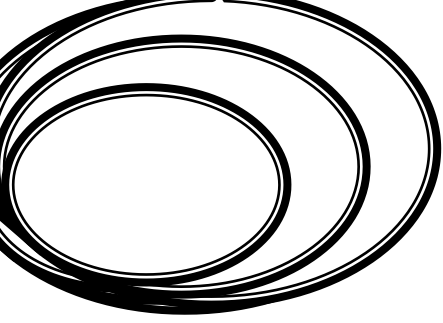
Au fur et à mesure que les environnements croissent en taille et en diversité, et que leur sécurisation machine par machine devient plus difficile, de plus en plus de sites se tournent vers un modèle de sécurité par réseau. Dans ce cas, vous vous concentrez sur le contrôle de l'accès réseau aux divers serveurs et aux services qu'ils offrent au lieu de les sécuriser un par un. Les approches de sécurité réseau comprennent la réalisation de firewalls qui protègent les systèmes et les réseaux internes en utilisant des principes d'authentification puissants (comme le mot de passe à utilisation unique), ainsi que le chiffrement des données particulièrement sensibles pendant qu'elles traversent le réseau.

Conclusion

Nous voulions dans ce chapitre donner une vue d'ensemble des risques et menaces qui existent dans le domaine informatique. Notre but était de sensibiliser avant tout, car beaucoup d'incidents sont dus à une mauvaise information.

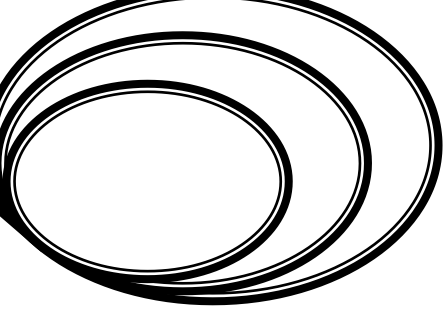
On a conclu, aucun modèle de sécurité ne peut pallier les problèmes d'encadrement, la sécurité informatique n'empêchera pas votre personnel de perdre du temps, de se chercher mutuellement des poux dans tête ou de vous gêner.

Alors, sécuriser le système d'information est plus en plus difficile, surtout à l'heure où le nombre d'applications et degré d'ouverture vers l'extérieur vont croissant. Définir les personnes autorisées à accéder au système d'information constitue l'une des bases de la sécurité, pour atteindre ce degré de protection dans les organisations, il est nécessaire de mettre un filtrage d'un trafic basé sur une architecture de type Netfilter à l'aide d'Iptables qui fera l'objet du chapitre suivant.



Chapitre 2 :

Sécurité des réseaux informatiques



Chapitre 3 :

Les outils de configuration d'un firewall

III.1 Introduction

Le filtrage d'un trafic au sein d'une organisation consiste d'examiner les paquets entrant aux réseaux et prendre des décisions sur le traitement à leur appliquer. C'est ce qui est appliqué avec un firewall, afin d'implémenter ces règles il faudra utiliser Netfilter à l'aide d'Iptables et présente linux et la distribution.

Ce chapitre consiste à étudier le firewall Netfilter de linux, son architecture, et ces tables et notamment sa commande d'implémentation Iptables.

III.2 Définition Linux

Linux est un système d'exploitation de type **Unix, libre et ouvert**.

Il est distribué sous la licence **GPL** (General Public licence de la Free Software Foundation). Cette licence a pour but de protéger les droits des développeurs, tout en permettant une diffusion et une utilisation totalement libre du logiciel, ainsi que du code source. Le code source doit d'ailleurs accompagner les versions binaires, au moins être rendu disponible. Les utilisateurs peuvent, à leur guise, modifier le code source et redistribuer cette nouvelle version, à la condition qu'elle soit elle-même sous licence GPL [13].

III.2.1 Debian GNU/LINUX

Debian est une organisation à but non lucratif constituée d'un millier de développeurs bénévoles répartis sur toute la planète. Elle est dirigée par un Project Leader élu par les développeurs. Les décisions se prennent au consensus ou par vote.

Simple d'utilisation, Debian n'offre pas d'autres supports que celui de la communauté Open Source, ce qui est parfois considéré comme un handicap comparé aux distributions fournies par des entreprises ajoutant ce type de service [13].

III.2.3 Avantages de système linux [13]

- **Open source (libre)**, le code source du noyau système et des programmes sont accessibles à tous (majoritairement sous la licence GPL (licence de public général)).
- **Gratuit (aucune licence à payer)** donc des économies pour les entreprises. Ils peuvent alors faire plus de bénéfices sur certains services comme l'installation et le support Linux. Certaines distributions Linux sont payantes mais la plupart d'entre elles sont moins chères que Windows.

- **Compatibilité multi-architecturales**, Linux s'installe sur toute sorte de machines : Intel (x86), powerpc, sparc, amd64 (ia64), mips, alpha, arm,...
- Il y a **plus de supports techniques et de documentations gratuites** sous Linux que dans Windows et ce, dans toutes les langues connues. Les réponses données dans les forums Linux en général sont techniquement "plus souples", "plus pointues et détaillées" dans le cas des problèmes sérieux.
- **Les bogues critiques sont corrigés rapidement** car les programmes concernés sont dits ouverts (open source). Plus il y aura des yeux regardants le code d'un logiciel et moins il y aura de bogues à court terme.

III.3 Squid

Un serveur **Squid** est un serveur mandataire (*proxy*) et un reverse proxy capable d'utiliser les protocoles FTP, HTTP, et HTTPS. Contrairement aux serveurs proxy classiques, un serveur Squid gère toutes les requêtes en un seul processus d'entrée/sortie, non bloquant, C'est un logiciel libre distribué sous licence GNU GPL [10].

III.3.1 Le logiciel

Squid, principal composant de ce système, assure les fonctions de [10]:

- Cache, pour optimiser la bande passante ;
- identification des utilisateurs, nous en verrons une simpliste et une nettement plus complexe ;
- filtrage d'accès « basique ».

III.3.2 Principe de fonctionnement

Squid tourne en tâche de fond (daemon). Il écoute sur un port spécifique (3128 par défaut, mais il est possible d'utiliser 8080, plus habituel pour un proxy HTTP). L'éventuel module d'identification vient se greffer dessus, ce qui fait apparaitre un certain nombre de processus fils (5 par défaut). Au total, une fois Squid configuré, il n'y aura qu'à démarrer Squid et les processus d'identification et de filtrage avancé démarreront avec. [8] [12].

III.3.3 Objectif

Installer un système de proxy cache pour HTTP. Ce proxy-cache propose deux fonctions principales [12]:

- L'optimisation de la bande passante sur le lien Internet, lorsque de nombreux clients sont connectés et qu'ils visitent plus ou moins les mêmes sites, à la condition, bien sûr, que ces sites ne soient pas trop dynamiques, ASP, JSP, PHP... ni chiffrés (HTTPS). la fonction cache présente de moins en moins d'intérêt. Il en reste un cependant, surtout pour les illustrations qui ne sont pas encore toutes en flash ;
- le contrôle et le filtrage de l'accès à la toile, en se servant des URI et, éventuellement, des noms d'utilisateurs, si l'on fait de l'authentification de ces derniers, autant de choses qu'il est difficile, voire impossible de faire avec du filtrage de paquets. En effet, en travaillant au niveau du protocole HTTP, il devient possible de mettre en place des filtres d'URI, des analyses de contenu dans les documents, alors qu'au niveau IP, nous ne pourrions filtrer que sur des adresses et des ports.

Tout responsable d'un réseau local à l'usage de mineurs et connecté à l'Internet se doit de mettre en place un tel système de filtrage de manière à éviter, autant que possible, l'accès à des sites que la morale réproouve, d'autant qu'il s'agit d'une obligation légale.

III.4 Netfilter

Est le nom d'une partie du kernel linux qui est destinée à assurer la surveillance de tous les transferts de données réseaux. Sa tâche est de faire du « Network packet filtering », c'est-à-dire du « filtrage de paquets Réseaux ».

Il se place entre la couche réseau du kernel Linux, et la couche applicative.

Comme Netfilter est un élément implanté profondément dans le kernel Linux (le « kernel space » ou « l'espace du cœur » en français), l'unique moyen que nous ayons de dialoguer avec lui est un programme appelé « Iptables » [11].

III.5 Architecture Netfilter

En tout état de cause, quelles que l'origine et la destination des paquets, ils vont entrer dans la pile de protocoles IP par le même point et en sortir par le même autre point.

Netfilter se présente comme une série de 5 « hooks » (points d'accrochage) sur lesquels des modules de traitement des paquets vont se greffer, ces points sont [11].

- PRE_ROUTING
- IN_PUT
- FORWARD
- POSTROUTING
- OUT_PUT

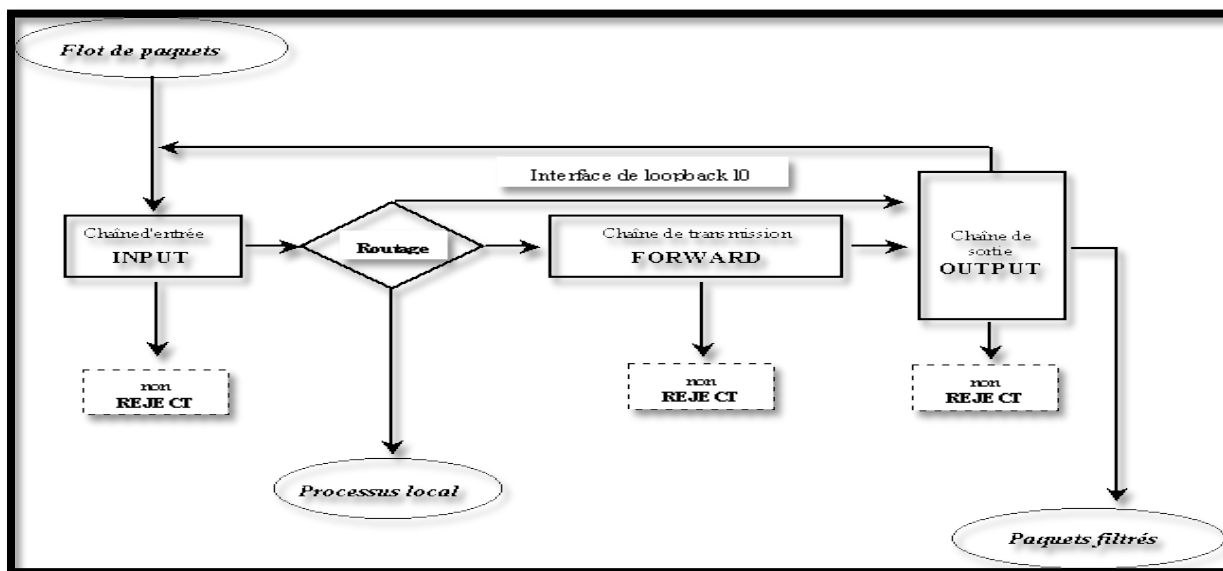


Figure III.1 : Architecture de Netfilter

III.6 Fonctionnement

Pour opérer le filtrage de paquets, Netfilter stocke un ensemble de règles définies par l'utilisateur. Ces règles sont enregistrées dans des tables sous formes de chaînes. Lorsque Netfilter doit traiter un paquet il applique l'ensemble des règles d'une chaîne les une à la suite des autres. Si le paquet correspond aux critères définis par la règle alors l'action associée à la règle (cible) est effectuée [14].

III.6.1 Les tables, chaînes et cibles

Aux points d'accès sont associées des chaînes de traitement. C'est dans celles-ci que sont effectués les tests. Elles sont regroupées par tables selon le type de traitement.

Les tables sont ajoutées par des modules. Il en existe 3 principales pouvant être utilisées [12] [14] :

Voici ci-après leurs noms et la tâche à laquelle elles sont destinées.

Table	Description
Filtre	Cette table permet de filtrer les paquets. Typiquement ce sera pour les accepter ou non.
Nat	Avec cette table on peut réaliser des translations d'adresse (ou de ports). Ceci sera notamment utile pour partager une connexion.
Mangle	Elle sert pour modifier les en-têtes des paquets. On la rencontrera parfois pour marquer des paquets afin que d'autres applications puissent les reconnaître.

Tab III.1 : Tables principales de Netfilter

A l'intérieur d'une table on peut trouver plusieurs chaînes. Ce sont elles qui contiendront les règles à appliquer aux paquets. Ces règles seront évaluées séquentiellement. On trouve deux types de chaînes.

Tous d'abord celles qui sont associées aux différents points d'entrées existants. Un paquet atteignant un de ces points sera envoyé vers la chaîne associée. Ce sont les fonctions de rappel évoquées précédemment qui réalisent cela. Elles effectuent les uns après les autres les tests qui contiennent la chaîne. Ces chaînes sont en nombre fini et ne sont pertinentes que pour certaines tables. Le tableau suivant les liste en indiquant quelle table a une chaîne de ce type.

Chaîne	Table	Description
PREROUTING	Nat, Mangle	Par cette chaîne passeront les paquets entrant dans la machine avant routage.
INPUT	Filter	Cette chaîne traitera les paquets entrants avant qu'ils ne soient passés aux couches supérieures (les applications).
FORWARD	Filter	Ce sont les paquets uniquement transmis par la machine sans que les applications n'en aient connaissance.
OUTPUT	Filter Nat, Mangle	Cette chaîne sera appelée pour des paquets envoyés par des programmes présents sur la machine.
POSTROUTING	Nat	Les paquets prêts à être envoyés (soit transmis, soit générés) seront pris en charge par cette chaîne.

Tab III.2 : chaînes de Netfilter

La cible s'agit du traitement que l'on décide d'appliquer au paquet. C'est la cible qui se chargera de faire les opérations nécessaires. En plus de celles prédéfinies il est possible

d'indiquer comme cible une chaîne utilisateur. Cela permet d'imbriquer différents tests et traitements. Chaque chaîne peut être vue comme un ensemble de tests

Cible	Description
ACCEPT	Les paquets envoyés vers cette cible seront tout simplement acceptés et pourront poursuivre leur cheminement au travers des couches réseaux.
DROP	Cette cible permet de jeter des paquets qui seront ignorés.
REJECT	Permet d'envoyer une réponse à l'émetteur pour lui signaler que son paquet a été refusé.
LOG	Demande au noyau d'enregistrer des informations sur le paquet courant. Cela se fera généralement dans le fichier /var/log/messages (selon la configuration du programme syslogd).
MASQUERADE	Cible valable uniquement dans la chaîne POSTROUTING de la table Nat. Elle change l'adresse IP de l'émetteur par celle courante de la machine pour l'interface spécifiée. Cela permet de masquer des machines et de faire par exemple du partage de connexion.
SNAT	Egalement valable pour la chaîne POSTROUTING de la table Nat seulement. Elle modifie aussi la valeur de l'adresse IP de l'émetteur en la remplaçant par la valeur fixe spécifiée.
DNAT	Valable uniquement pour les chaîne PREROUTING et OUTPUT de la table Nat. Elle modifie la valeur de l'adresse IP du destinataire en la remplaçant par la valeur fixe spécifiée.

Tab III.3 : cibles de Netfilter

III.7 Présentation d'Iptables

Iptables est donc une commande qui seul le root peut lancer. Selon but est de dialoguer avec Netfilter afin de contrôler les **règles** des **chaînes** dans le but de configurer les **tables** [14][15].

Iptables est la boîte à tout faire de Netfilter. Cette commande va pouvoir :

- ❖ **Rajouter** des règles/chaînes.
- ❖ **Supprimer** des règles/ chaînes.
- ❖ **Modifier** des règles/chaînes.

❖ **Afficher** les règles/chaînes.

« Iptables » est un programme qui lance en ligne de commande et qui attend de nombreux paramètres.

Option	Rôle
-L	Affiche toutes les règles de la chaîne indiquée.
-F	Supprime toutes les règles de la chaîne. Si aucune chaîne n'est spécifiée, toutes celles de la table sont vidées.
-N	Crée la chaîne utilisateur avec le nom passé en paramètre.
-X	Supprime la chaîne utilisateur. si aucun nom n'est spécifié, toutes les chaînes utilisateur seront supprimées.
-P	Modifie la politique par défaut de la chaîne. Il faut indiquer en plus comme paramètre la cible à utiliser.
-A	Ajoute une règle à la fin de la chaîne spécifiée.
-I	Insère la règle avant celle indiquée. Cette place est précisée par un numéro qui fait suite au nom de la chaîne. La première porte le numéro 1. Si aucun numéro n'est indiqué la règle est insérée au début.
-D	Supprime une règle de la chaîne. Soit un numéro peut être précisé, soit la définition de la chaîne à supprimer (ses tests de concordance et sa cible).
-R	remplace une règle dans la chaîne (Replace)
-o	interface de sortie (output)
-t	table (par défaut contenant les chaînes INPUT, FORWARD, OUTPUT)
-i	interface d'entrée (input)
-j	règle à appliquer (Jump)
-lo	localhost (ou 127.0.0.1, machine locale)

Tab III.4 : Les options d'Iptables

III.7.1 Le suivi de connexion (connection Trackers)

Le suivi de connexion est un concept essentiel dans Netfilter. 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 faut à un moment donné, dire quelques mots à propos de ce suivi de connexion [11] [15] :

Principe: Etat du paquet. Une liste de plusieurs valeurs peut être indiquée en les séparant par des virgules. L'état de ce paquet est comparé alors à ces valeurs.

NEW correspond à un paquet initiant une nouvelle connexion.

ESTABLISHED est un paquet participant à une conversation déjà établie.

RELATED est pur un paquet qui ouvre une nouvelle connexion, mais ceci en rapport avec une précédente déjà établie.

INVALID indique un paquet qui n'est rattaché à aucune connexion.

Ces états peuvent être utilisés avec la correspondance **state** pour sélectionner les paquets à partir de leur état de traçage de connexion. C'est ce qui rend la machine d'état si puissante et efficace pour votre pare-feu.

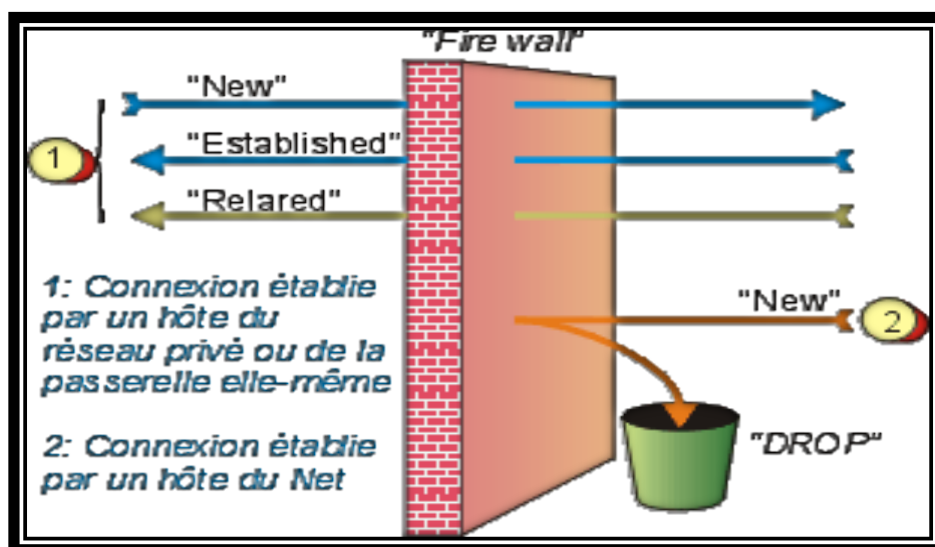


Figure III.2 : connection Trackers

Cette figure montre le fonctionnement du suivi de connexion. Les différents états dont les quels être un trafic traversant le firewall.

Conclusion

On a présenté dans ce chapitre, le firewall Netfilter de Linux, son fonctionnement son architecteur ainsi nous avons présenté les différentes commandes Iptables.

Dans le chapitre suivant on va donner les différentes étapes de configuration d'un firewall en utilisant les règles d'Iptables.

IV.1 Introduction

Ce chapitre est consacré à la description détaillée comment utiliser Netfilter pour configurer un puissant pare-feu avec suivi de connexion sous linux : les étapes de développement du (firewall authentifiant, squid,...).

Ce chapitre est suivi par une annexe qui décrit la conception de la topologie du firewall.

IV.2 Le but

Dans ce chapitre, nous allons bâtir un pare-feu avec suivi de connexion pour Linux. Notre firewall va s'exécuter sur un ordinateur portable, de bureau, serveur ou routeur sous Linux , son but principal est d'autoriser seulement certains types de trafic réseau à le traverser. Pour augmenter la sécurité, nous allons configurer le firewall pour ignorer ou rejeter le trafic qui ne nous intéresse pas ainsi que le trafic potentiellement dangereux pour la sécurité.

Pour configurer le firewall sous proxy sous linux il faut faire :

IV.3 configuration avec IP statique

Pour qu'il soit facile à repérer un serveur doit avoir une adresse IP statique, nous avons configuré cette adresse dans le principal fichier de configuration du réseau « `/etc/network/interfaces` », après l'ouverture de ce fichier avec l'éditeur de textes « nano » par la commande :

```
# nano /etc/network/interfaces
```

Nous avons modifié le fichier comme site :

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address xxx.xxx.xxx.xxx
netmask xxx.xxx.xxx.xxx
```

Remarque : après chaque configuration de la carte réseau, il faut la redémarrer par la Commande:

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

IV.4 Serveur DHCP

Nous avons mis en place un serveur dhcp, pour attribuer des adresses ip dynamique au client.

Nous avons installé un outil qui s'appelle « isc-dhcp-server » par la commande apt-get :

```
#apt-get install isc-dhcp-server
```

Ensuite on a édité le fichier /etc/dhcp/dhcpd.conf par l'éditeur nano et nous avons modifié des lignes comme suite :

```
Option domain-name "univ-tiaret.dz" ;
```

Et ajouter les lignes suivantes :

```
#déclarer l'IP de reseau et le mask de reseau  
Subnet 192.168.0.0 netmask 255.255.0.0{  
  
# declaration de la plage d'adresse IP  
Range 192.168.0.1 192.168.255.254 ;  
  
#Déclaration de la passerelle  
Option routers 192.168.1.1 ;  
  
}
```

Enfin, on modifie le fichier « isc-dhcp-server » qui se trouve dans le répertoire « /etc/default/ » pour ajouter l'interface d'écoute « eth1 » pour le serveur DHCP :

```
INTERFACES="eth1"
```

eth1 c'est l'interface où les clients sont connectés.

IV.5 Installation Squid

Squid, principal composant de ce système, assure les fonctions de :

- Cache, identification des utilisateurs, filtrage d'accès.

Pour installer le Squid on utilise la commande suivante :

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

Comme vous le voyez, on l'installe et il démarre tout seul.

Après le redémarrage on obtient l'interface suivante :

```
# ps aux | grep [s]quid
```

```
root 571 0.0 1.8 3824 1124 ? S 16:26 0:00 /usr/sbin/squid -D -sYC
```

```
proxy 574 0.8 8.2 8468 5068 ? S 16:26 0:03 (squid) -D -sYC
```

Effectivement, il tourne. N'y aurait-il rien de plus à faire ? Vérifions tout de suite. Squid utilise par défaut le port 3128. Configurons donc un navigateur du LAN pour l'utiliser et essayons un URI au hasard...

Résultat avant la configuration de squid :

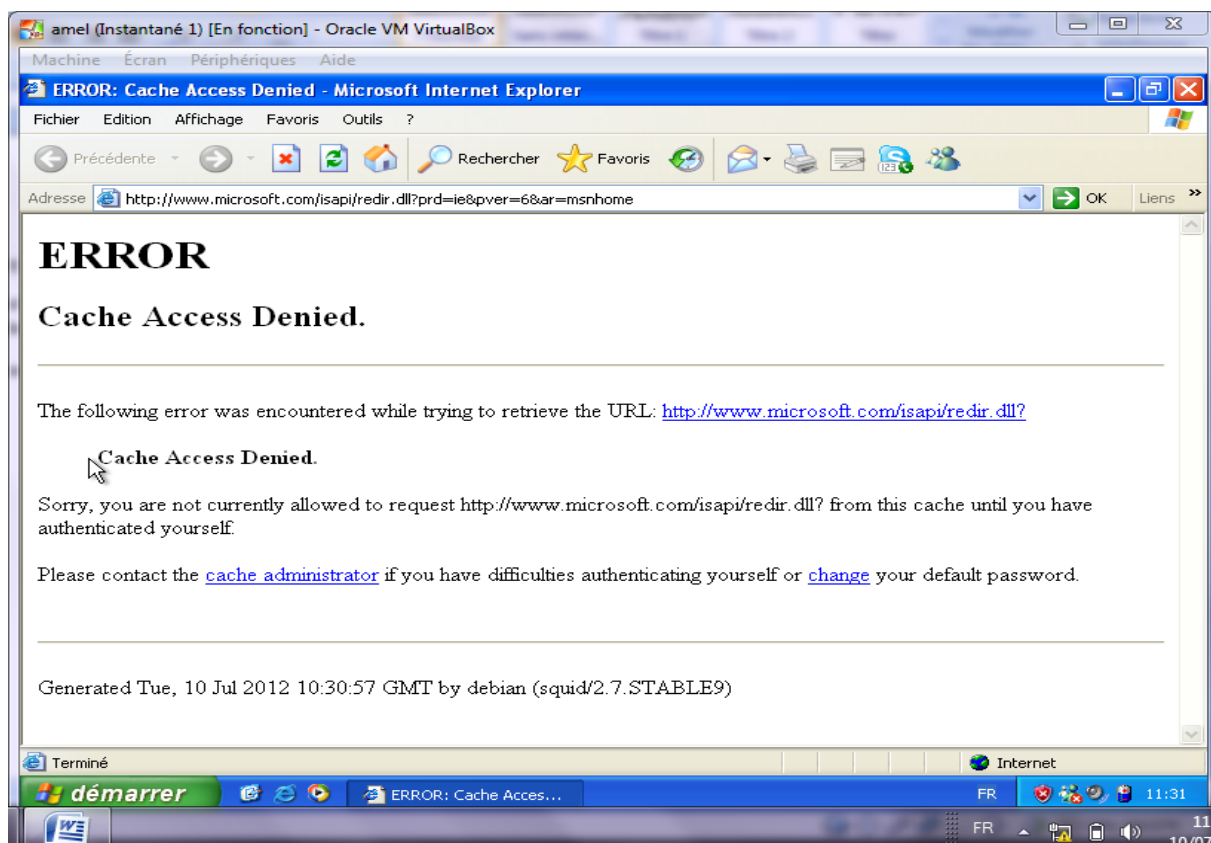


Figure IV.1 : Interface de web des erreurs d'authentification d'un client

Pour d'accès a la page web d'internet il faut créés les ACL représentant le LAN comme suite :

IV.5.1 Configuration minimale

Les ACL (Access Control Lists) permettent de définir des conditions sur les IPs, les ports, le contenu de certains textes.

Le fichier de configuration est : **/etc/squid/squid.conf**.

```
acl manager proto cache_object

acl localhost src 127.0.0.1/255.255.255.255

acl to_localhost dst 127.0.0.0/8

acl SSL_ports port 443

acl Safe_ports port 80          # http
acl Safe_ports port 21         # ftp
acl Safe_ports port 443       # https
acl Safe_ports port 70        # gopher
acl Safe_ports port 210       # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280       # http-mgmt
acl Safe_ports port 488       # gss-http
acl Safe_ports port 591       # filemaker
acl Safe_ports port 777       # multiling http

acl CONNECT method CONNECT

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

http_access allow localhost

http_access deny all

icp_access allow all
```

```
http_port 3128
hierarchy_stoplist cgi-bin ?
access_log /var/log/squid3/access.log squid
acl QUERY urlpath_regex cgi-bin \?
cache deny QUERY
refresh_pattern ^ftp:          1440      20%      10080
refresh_pattern ^gopher:      1440      0%       1440
refresh_pattern .              0         20%     4320
icp_port 3130
coredump_dir /var/spool/squid3
```

Dans un premier temps, disons pour aller très vite au but, que :

- les « acl » (Access Control List) permettent de définir, par exemple, un plage d'adresses IP, celles qui constituent notre réseau local ;
- les « http_access » (restrictions) qui définissent l'autorisation ou l'interdit, pour une acl donnée.

Les restrictions indiquent quoi faire lorsque ces conditions sont vérifiées. On autorise ou on interdit en fonction d'une ACL ou d'un groupe d'ACLs, le sens de « restriction » est donc à prendre avec un peu de recul, une restriction pouvant être une autorisation. La première restriction vérifiée est la bonne, d'où l'importance de l'ordre dans lequel elles sont placées.

Sans faire une analyse détaillée, nous voyons que dans la configuration par défaut, seul « localhost » peut utiliser le proxy (Allow localhost). Si cette condition n'est pas respectée, la règle suivante étant deny all, personne ne passe. Il nous faut donc faire intervenir la notion de réseau local.

IV.5.2 Créer une ACL représentant le LAN

Bien entendu, l'idée de faire plutôt Allow all est une mauvaise idée. Si votre proxy a un pied dans l'Internet (s'il est installé sur la passerelle), vous risquez un proxy ouvert, avec tous les usages pervers que l'on peut en faire...

Modifions le fichier **squid.conf** en ajoutons les deux lignes suivantes :

```
Acl LocalNet src 192.168.0.0/16  
  
http_access allow LocalNet
```

Nous avons créé une ACL nommée LocalNet représentant notre réseau local (acl LocalNet src 192.168.0.0/16), et lui avons donné l'autorisation de passer le proxy (http_access allow LocalNet). Nous relançons squid :

```
# /etc/init.d/squid reload  
Rrloading Squid HTTP Proxy configuration files.
```

Après la configuration d'acl, les clients autorisés peuvent accéder au serveur comment il est indiqué dans la figure suivante :



Figure IV.2 : l'accès d'un client autorisé à une page web

Nous disposons d'un proxy cache en état de marche pour notre réseau local.

IV.6 Configuration de cache :

Installer un système de proxy cache pour HTTP.

```
cache_dir ufs /var/spool/squid 1024 256 256

# Les journaux

cache_access_log /var/log/squid/access.log common

cache_log /var/log/squid/cache.log

cache_store_log /var/log/squid/store.log

cache_swap_log /var/log/squid/cache_swap.log

emulate_httpd_log on
```

IV.7 Identifier les utilisateurs

Dans la configuration mise en œuvre jusqu'ici, nous ne faisons pas de contrôle sur les utilisateurs, seulement sur les IPs des machines clientes. Vous pouvez souhaiter identifier vos utilisateurs lorsqu'ils vont surfer sur le Net. Dans ce cas, il vous faudra mettre en place un système d'identification.

Il y a plusieurs méthodes disponibles pour authentifier les utilisateurs du proxy. Elles font tout appel à un programme extérieur, différent suivant le moyen choisi. Debian propose les modules suivants :

```
squid_ldap_auth, msnt_auth, ncsa_auth, pam_auth, sasl_auth, smb_auth, yp_auth, getpwnam_auth, ntlm_auth, digest_ldap_auth, digest_pw_auth...
```


- **ncsa_auth** qui permet d'identifier les utilisateurs à partir d'un fichier local de type « htpasswd » ;

Nous allons dans un premier temps essayer `ncsa_auth`, ce ne sera peut-être pas le plus utile, surtout si le réseau local est un domaine Microsoft Windows, mais c'est le plus simple à mettre en œuvre.

IV.7.1 Construire un fichier d'utilisateurs

Nous allons créer un fichier `/etc/squid/users`

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

Nous le remplissons ensuite avec la commande **htpasswd**, normalement fournie dans le paquet `apache-common`.

```
# htpasswd -b /etc/squid/users <nom de l'utilisateur> <mot de passe>
```

A répéter autant de fois que nécessaire avec des vrais noms d'utilisateurs et des vrais mots de passe...

Le fichier se remplit comme suit :

```
# cat /etc/squid/users
user1:kCNRmSqyu3kZRY
user2:r7HN1QByNv8K2
user3:AP7PV8PCGurVk
```

Notez que les mots de passe sont chiffrés.

Vérifions que ceci fonctionne, en lançant « à la main » le module d'authentification `/usr/lib/ncsa_auth`. Nous entrerons alors dans une boucle où il faudra entrer sur une ligne un nom d'utilisateur et son mot de passe, séparés par un espace :

```
# /usr/lib/squid/ncsa_auth /etc/squid/users
user1 pw1
```

```
OK
user2 pw2
OK
user3 pw3
OK
user password
ERR No such user
```

Le système répond par OK ou par ERR suivant que l'authentification réussit ou non.

IV.7.2 Configurer squid pour réclamer l'identification de vos utilisateurs

Nous devons commencer par fournir quelques directives de type **auth_param** :

```
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
```

program, indiquez le chemin du module `ncsa_auth`, suivi du chemin du fichier des utilisateurs, séparés par une espace.

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

realm, n'est rien d'autre qu'un texte qui apparaîtra dans la fenêtre de demande d'identification.

credentialsttl, durée de vie de l'identification. A condition bien sûr que le navigateur ne soit pas fermé avant.

Il nous faut maintenant créer une “acl” supplémentaire, pour obliger l'identification,

```
acl Users proxy_auth REQUIRED
```

Puis n'autoriser l'accès que si le client est dans notre réseau et que l'identification est réussie :

```
http_access allow LocalNet Users
```

cette fois-ci, il y est. Ca devrait donc fonctionner :

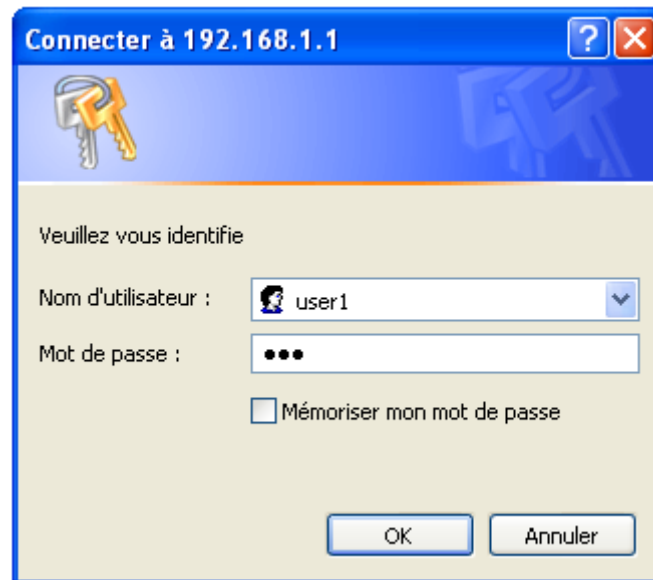


Figure IV.3 : demande d'authentification

IV.8 Étapes de configuration du firewall

Le coupe-feu Iptables est installé par défaut sur les machines Linux, il appartient aux outils de base de linux.

Les règles peuvent porter sur 3 chaînes :

- INPUT en entrée,
- FORWARD dans le cas d'un routage réseau,
- OUTPUT en sortie.

Nous allons créer un fichier appelé firewall dans le répertoire "/etc/init.d/ ", et on le remplisse par les nos règles d'Iptables suivantes :

```
# !/bin/sh
### BEGIN INIT INFO
# Provides:          Firewall maison
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:
# Default-Stop:
# X-Interactive:     false
# Short-Description: Firewall maison
### END INIT INFO
# Mise à 0
iptables -t filter -F
```

```
iptables -t filter -X
# On bloque tout
iptables -t filter -P INPUT DROP
iptables -t filter -P FORWARD DROP
iptables -t filter -P OUTPUT DROP
echo "Vidange des Tables : OK"
# Ne pas casser les connexions établies
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# Autorise le loopback (127.0.0.1)
iptables -t filter -A INPUT -i lo -j ACCEPT
iptables -t filter -A OUTPUT -o lo -j ACCEPT
echo "Loopback"
# ICMP (le ping)
iptables -t filter -A INPUT -p icmp -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -j ACCEPT
echo "Ping ok"
# DNS In/Out
iptables -t filter -A OUTPUT -p tcp --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -p udp --dport 53 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -t filter -A INPUT -p udp --dport 53 -j ACCEPT
echo "dns ok"
# HTTP + HTTPS Out
iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT
# HTTP + HTTPS In
iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 443 -j ACCEPT
echo "http ok"
# FTP Out
iptables -t filter -A OUTPUT -p tcp --dport 21 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 20 -j ACCEPT
# FTP In
# imodprobe ip_contrack_ftp # ligne facultative avec les serveurs OVH
iptables -t filter -A INPUT -p tcp --dport 20 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 21 -j ACCEPT
iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
echo "ftp ok"
# Mail SMTP:25
iptables -t filter -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 25 -j ACCEPT
# Mail POP3:110
iptables -t filter -A INPUT -p tcp --dport 110 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 110 -j ACCEPT
# Mail IMAP:143
iptables -t filter -A INPUT -p tcp --dport 143 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 143 -j ACCEPT
# Mail POP3S:995
iptables -t filter -A INPUT -p tcp --dport 995 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 995 -j ACCEPT
echo "mail ok"
# DHCP 67:68
iptables -A INPUT -p UDP -i eth1 --dport 67:68 --sport 67:68 -j ACCEPT
iptables -A OUTPUT -p UDP -o eth1 --sport 67:68 --dport 67:68 -j ACCEPT
echo "DHCP OK"
# Proxy squid:3128
```

```
iptables -A INPUT -p TCP -i eth1 --dport 3128 -j ACCEPT
iptables -A OUTPUT -p TCP -o eth1 --sport 3128 -j ACCEPT
echo "SQUID OK"
```

Après que nous avons introduit les règles d'Iptables dans ce fichier , nous allons lui donner le droit d'exécution par la commande suivante :

```
#chmod +x firewall
```

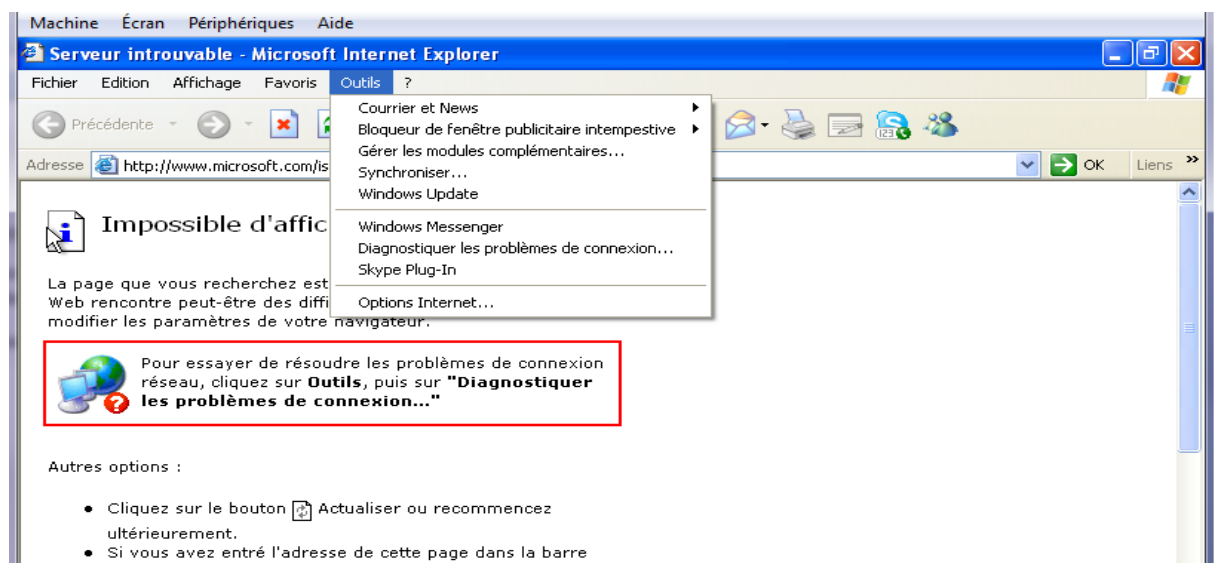
Enfin, on l'ajoute au processus de démarrage par la commande :

```
# update-rc.d firewall defaults
```

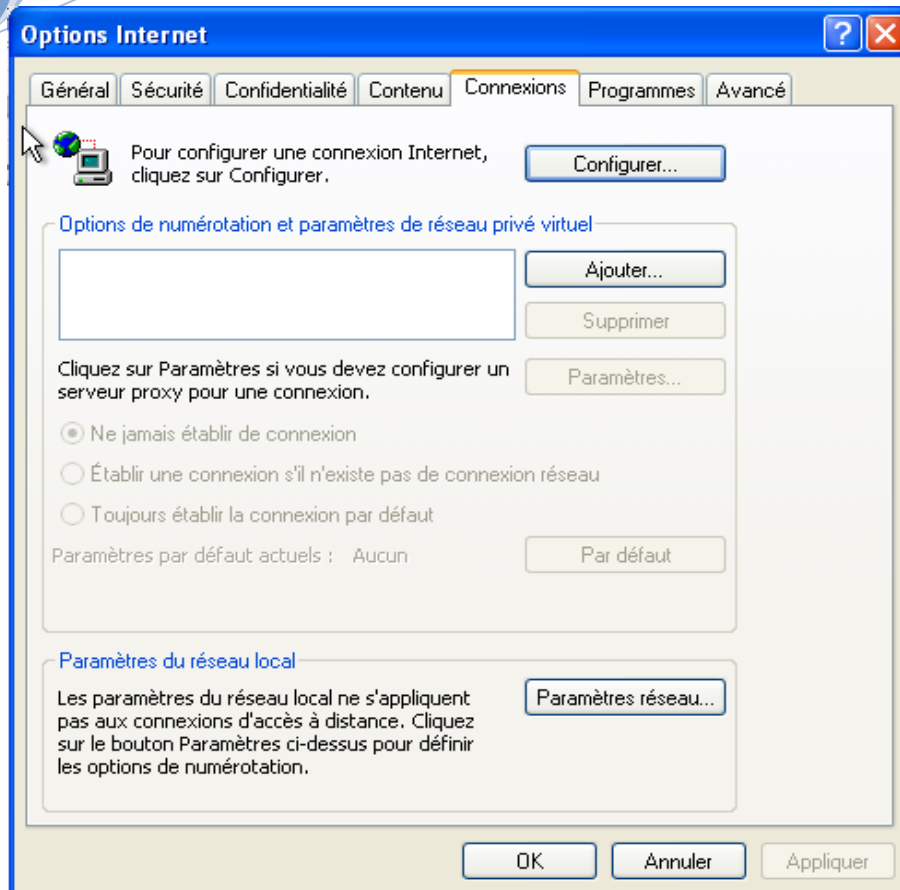
IV.9 Configuration proxy cote client:

Il faut que nous ajoutons pour chaque client l'@ IP de notre serveur proxy, pour y faire il faut procéder comme suite :

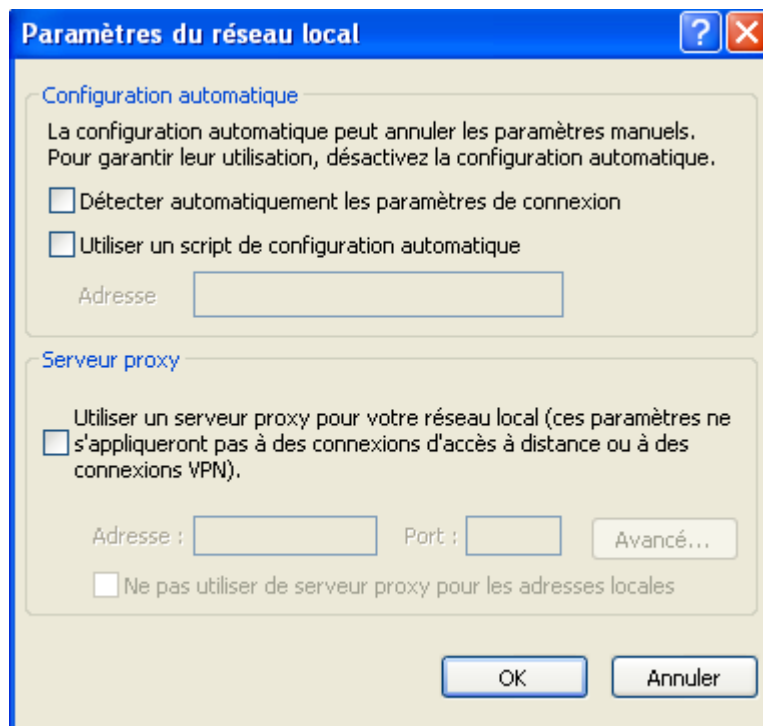
- Ouvrir un navigateur web au niveau de client



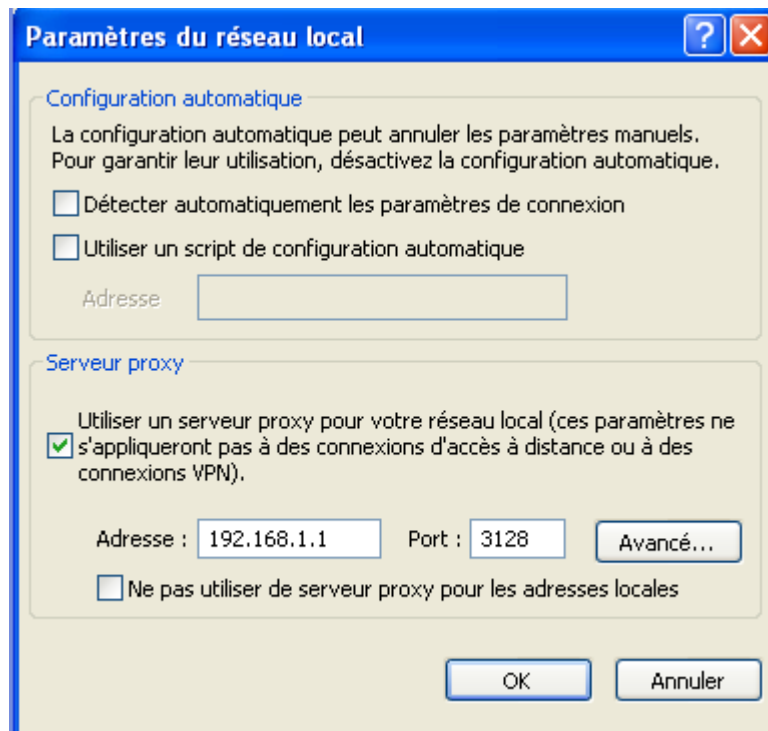
- Dans la barre menu, cliquer sur « outils » puis « option internet »



- Dans l'onglet « connexions », cliquer sur le bouton « Paramètres réseau... »



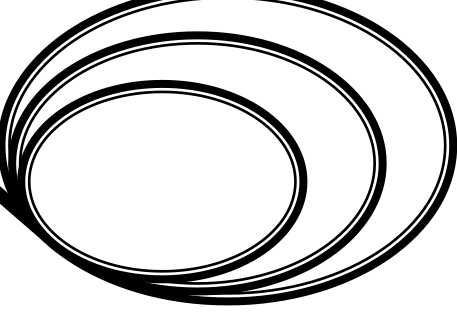
- Dans cette fenêtre, cocher « Utiliser un serveur proxy... connexions VPN) » et ajouter l'adresse IP de serveur « 192.168.1.1 » et le port « 3128 ».



- Valider par le bouton OK.

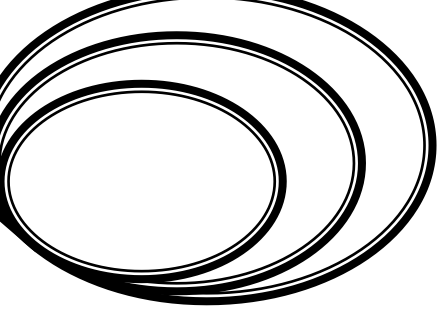
Conclusion

Dans ce chapitre on essaye de détaillé comment on réalise un pare-feu sous proxy sous linux.



Chapitre 4 :

Configuration d'un firewall



Annexe :

Mise en œuvre de la topologie réseau

1. Étapes de développement de la topologie

Notre topologie a été développée en utilisant les moyens de virtualisation. Dans le cadre de ce projet on a choisi (Virtual Box) téléchargeable à partir de site (www.Virtual-box.org), celui-ci permet de créer des machines virtuelles sous un système d'exploitation de choix qui est dans notre cas (linux / Debian) comme il offre aussi la possibilité de mettre ces machines virtuelles en réseau local.

1.1 Topologie du réseau à construire

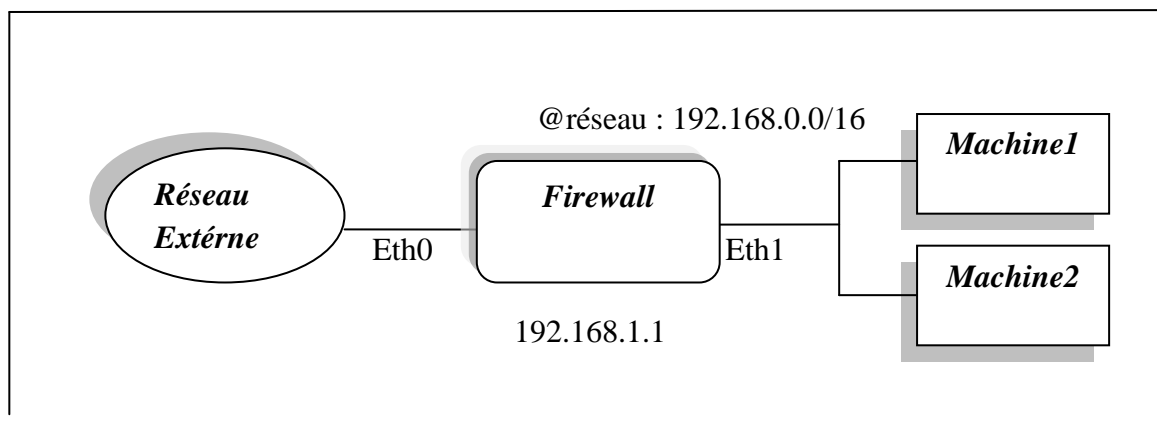


Figure 1 : topologie du réseau

1.2 Création des machines virtuelles

Pour construire notre topologie on a besoin de créer au minimum deux machines, la première sert au développement du firewall, la deuxième pour représenter un client, mais on peut créer autant de machines clientes selon notre besoin, ces machines fonctionnent avec un OS (linux/Debian), comme elles sont représentées dans la figure suivante :

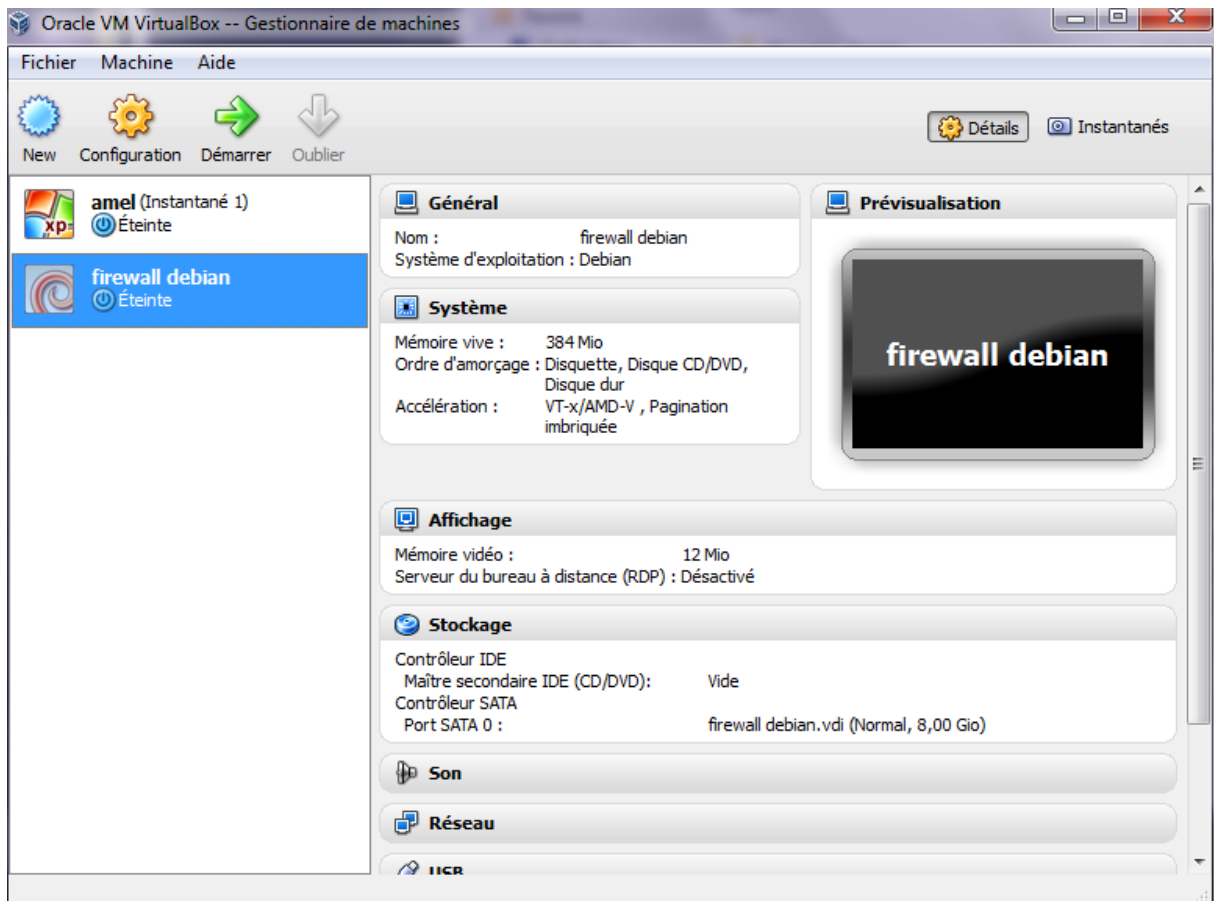


Figure 2 : les machines virtuelles

1.3 Construction du Réseau Local

Pour avoir une communication entre ces machines alors on a eu besoin de les interconnectés en réseau local à l'aide de Virtual _ Box et de faire aussi un adressage statique, en suivant les étapes ci-dessous :

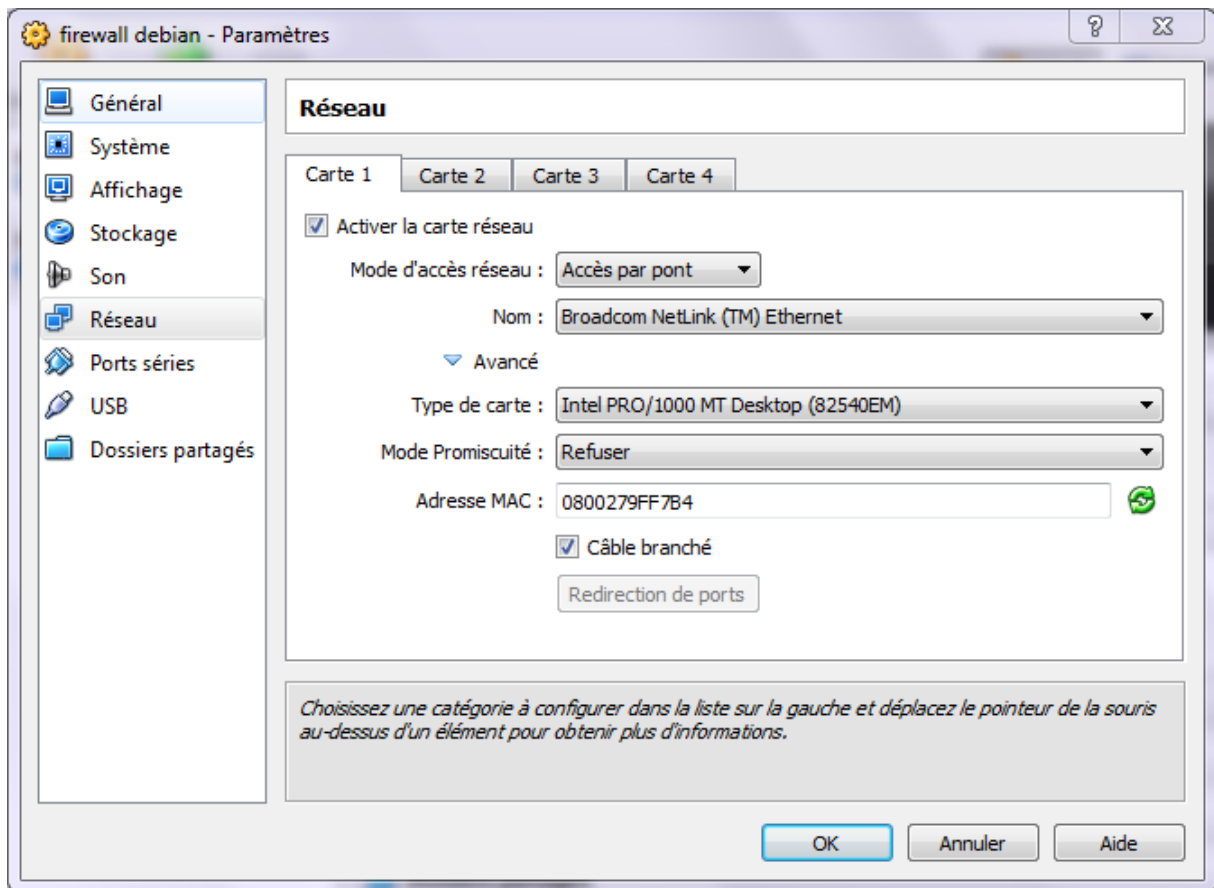


Figure 3 : Attachement de l'interface de connexion du client

Cette figure montre les propriétés de la machine firewall qui a deux interfaces :

La première est attachée à un réseau externe (Carte accès par pont), son adresse est obtenue à partir d'un serveur DHCP, c'est donc une adresse qui s'obtient d'une manière dynamique. La deuxième (par feu) est attachée à un réseau local avec une adresse statique comme on peut le voir sur la figure suivante :

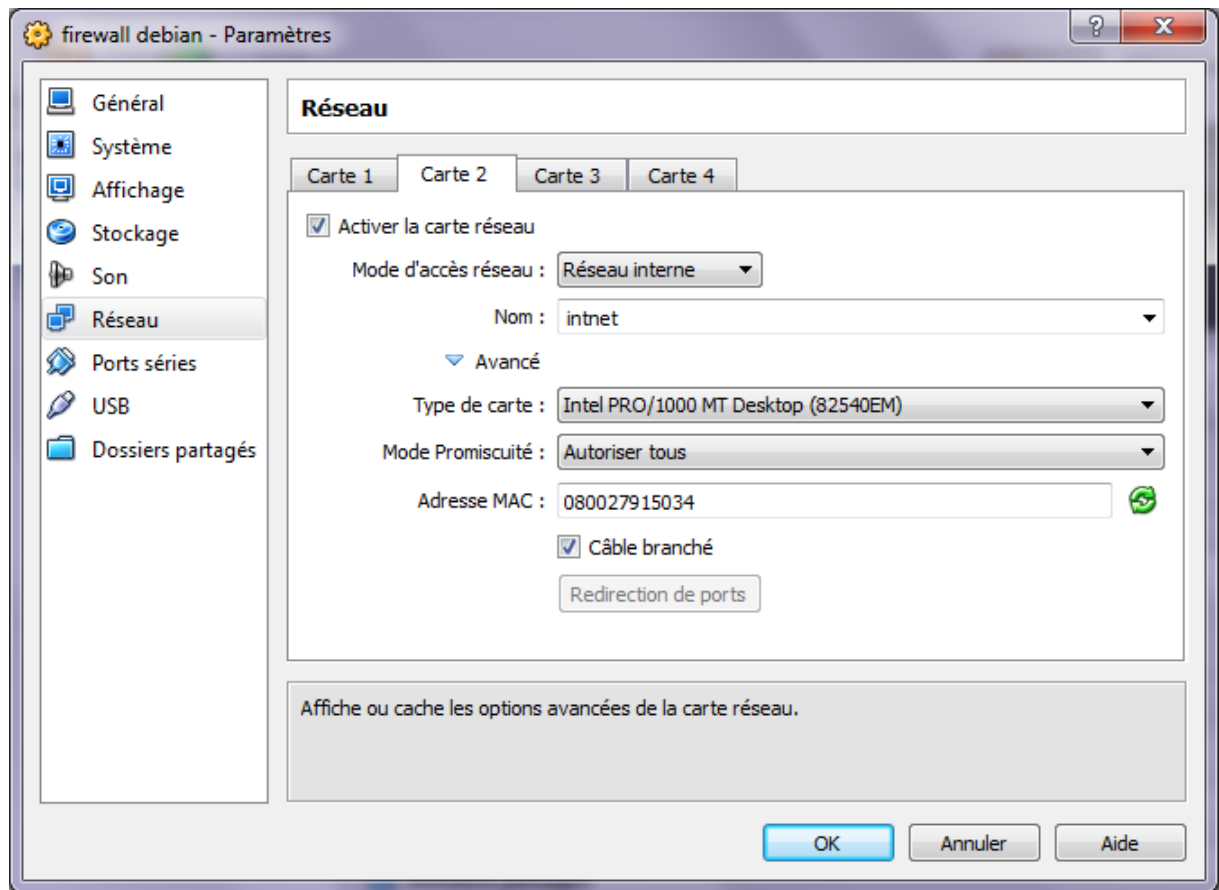
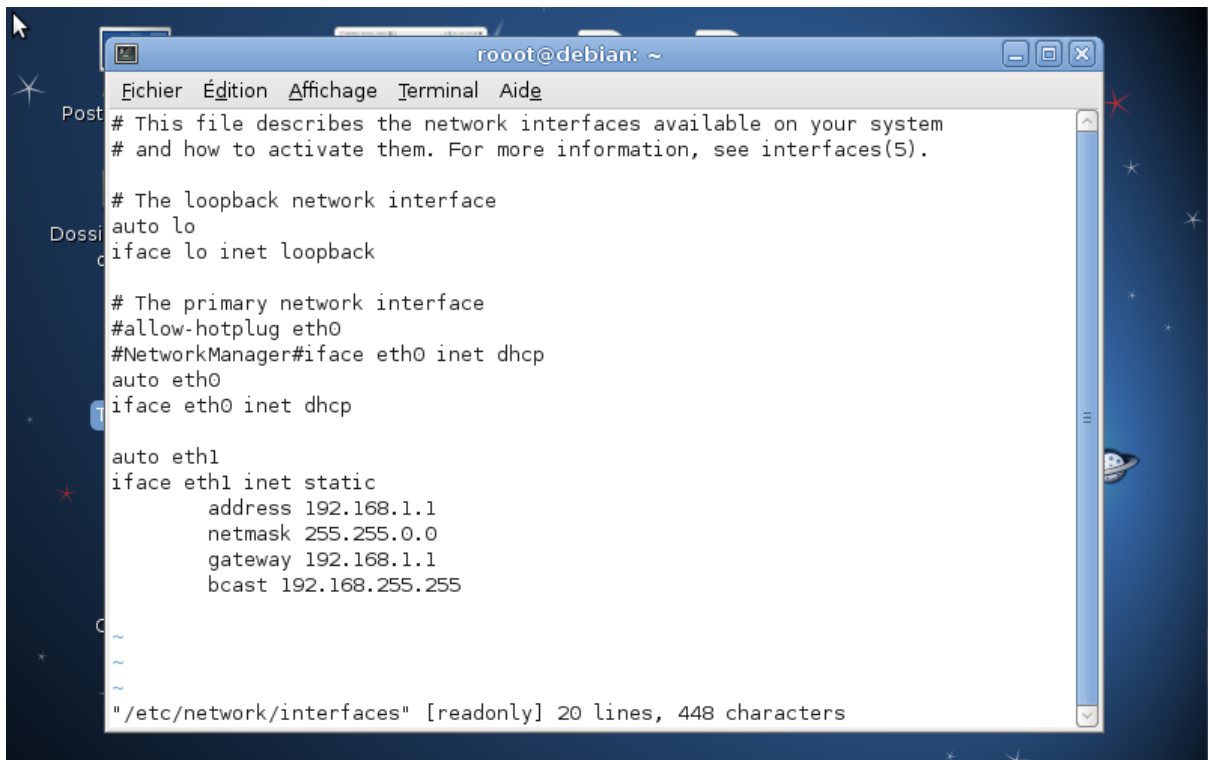


Figure 4 : Attachement de l'interface de connexion du par feu

1.4 Affectation des adresses

En modifiant le fichier : vi /etc/network/interfaces

A screenshot of a terminal window on a Debian system. The window title is 'root@debian: ~'. The terminal shows the contents of the file /etc/network/interfaces. The text is as follows:

```
Fichier Édition Affichage Terminal Aide
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
#NetworkManager#iface eth0 inet dhcp
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.1.1
    netmask 255.255.0.0
    gateway 192.168.1.1
    bcast 192.168.255.255

~/
~/
~/
"/etc/network/interfaces" [readonly] 20 lines, 448 characters
```

Pour interconnecté les autres machines au réseau local on utilise la même procédure que nous avons utilisé pour interconnecté la deuxième interface de la machine par feu en utilisant un adressage statique.

Afin d'avoir une communication entre ces machines, on active le forwarding avec la commande suivante :

```
echo 1 > /proc/sys/net/ipv4/forwarding.
```

On tape cette commande au niveau de la machine firewall car elle est responsable de l'acheminement des paquets de la machine émettrice à la machine réceptrice.

Bibliographies

Livres:

[1]-Building Internet Firewall, D.B Chapman and E. Zwicky, o'railly,1995

[2]-Internet Firewall and Network Security, New Riders publishing, 1995

[3]-firewall, Ferguson Niels, Schuneier Bruce, Cryptographie- En pratique, Vuibert, 2004, ISBN 978-2-71-174820-4.

[4]-Appréhension de la sécurité pour un ingénieur du monde des systèmes distribué.

Anderson Ross, Security Engineering, A Guide to building dependable Distributed System, Wiley 2^{ème} edition, 2008, ISBN 978-0-006852-6.

[5]-Technologie de la sécurité des réseaux

Stallings William, Sécurité des réseaux, applications et standard, Vuibert,2002,ISBN 9782-71-178653-4.

Sites Internet :

[6]- <http://www.misfu.com/cours-fonctionnement-firewall.html> par Jérôme Henry (consulter avril-2012)

[7]- <http://www.neufsecurite.com/Neuf-Securite/firewall.html> présentation du firewall (consulter avril-2012)

[8]- <http://www.commentCaMarche.com>(consulter avril-2012)

[9]-<http://www.depinfo.mines.inpl-nancy.fr/Members/Lahmadi/cours.Firewall.pdf>par Abdelkader Lahladi LORIA-école des Mines de Nancy-janvier 2008(consulter avril-2012)

[10]-<http://www.fr.wikipedia.org/w/index.php?title.php?squid&oldid=80400398> (consulter avril-2012)

[11]- <http://www.irp.nain-t.net/doku.php/220squid:star>(consulter mai-2012)

[12]- <http://www.alcove.com> Benjamin Drieu. Politique de sécurité (consulter mai-2012)

[13]-<http://www.linux.com/site> officiel sur linux (consulter juin-2012)

[14]- <http://www.netfilter.org/site> officiel de Netfilter (consulter juin-2012)

[15]-<http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.txt>. Paul Russel and Herald Welte, "Netfilter Hacking How-to" (consulter juin-2012)