



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE**

**MINISTER DE L'ENSEGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**



UNIVERSITE ABOU BAKR BELAKID-TLEMCEN

**FACULTE DE SCIENCE
DEPARTEMENT D'INFORMATIQUE**

**Projet de fin d'étude
Pour l'obtention du
DIPLOME DE LICENCE
EN
INFORMATIQUE**

THEME

**Netfiltre et iptables appliqué sur un réseau d'une
machine virtuelle**

Présenté par :

M^{ll} Benmansour Radjaa

M^{ll} Benmansour Zineb

Soutenu en 27 juin 2013 devant le jury

Mm F.DIDI

EXAMINATEUR

Mm N.LABRAOUI

EXAMINATEUR

Mr S.BENAISSA

ENCADREUR

Année Universitaire : 2012/2013

Remerciements

Nous remercions dieu le tout puissant qui nous a donné durant toutes ces années la santé, Le courage et la foie en nous même pour pouvoir avancer et mener nos études à leurs termes.

Tous d'abord nous remercions notre encadreur Monsieur BENAISSA SAMIR d'avoir rempli parfaitement son rôle et pour les orientations précieuses dont il nous a fait part nous ne saurions le remercié assez pour son soutien et son suivi scientifique .Nous lui devons beaucoup pour sa confiance qu'il nous a témoigné et pour son encouragement et conseil qu'il nous a prodigué.

Nous remercions aussi tous nos enseignants du département d'Informatique et tous les enseignants qui ont participé à notre formation depuis la première année primaire.

Nos mots de reconnaissances vont à tous ceux qui ont contribué de prés ou de loin à L'élaboration de cette modeste étude.

Dédicaces

*Je dédicace ce modeste travail au gents qui mes chère a mon cœur
Mes parents qui mon soutenue durons tout ma vie. Que se soit dans
les grandes occasions ou petite, et je les remercie de leurs sacrifices
et leurs efforts au prés de mon éducation.*

Mes sœurs Souhila et Yasmine.

*A toute ma famille : grands parents paternelles et maternels, tantes et
oncles, mes cousins et cousines en souvenir de toutes les joies et les
forces qui nous unissent.*

A tous ceux qui m'aime, et que je leur manque fréquemment.

B. Radjaa

Dédicaces

*Je dédicace ce modeste travail au gents qui mes chère a mon cœur.
Mes parents qui mon soutenue durons toute ma vie. Que se soit dans
les grandes occasions ou petite, et je les remercie de leurs sacrifices
et leurs efforts au prés de mon éducation.*

*A mes chères ami(e)s, Mes sœurs Yasmin et Nerdjes et mon frère
Boumedyene, Et ma chère nièce Lilya .*

*A toute ma famille : grands parents paternelles et maternels, tantes et
oncles, mes cousins et cousines en souvenir de toutes les joies et les
forces qui nous unissent.*

A tous ceux qui m'aime, et que je leur manque fréquemment.

B.Zineb

Sommaire

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

Chapitre 1 : Introduction à la virtualisation

1. Introduction.....	2
2. Virtualisation	3
3. Les types de virtualisation	3
3.1. Hyperviseur de type 1	3
3.2. Hyperviseur de type 2	4
4. Avantages de la virtualisation	5
5. Inconvénients de la virtualisation	6
6. Virtualbox	7
7. Conclusion	7

Chapitre 2 : Notion de base d'un pare feu

1. Introduction	8
2. Firewall outil de protection	8
3. Technologies de filtrage	8
3.1. Le filtrage de paquet	8
3.2. Firewalls de niveau circuit	10
3.3. Firewalls de couche application	11
3.4. Filtrage dynamique de paquets	12
4. Architectures firewall	13
4.1. Firewall avec routeur de filtrage	13
4.2. Passerelle double ou réseau bastion	14
4.3. Firewall avec réseau de filtrage	14
4.4. Firewall avec sous-réseau de filtrage	15

5. Fonctionnement du pare-feu sous Linux :NetFilter/Iptables	16
5.1. Fonctionnement	17
5.2. Les tables	17
5.3. Les chaines	18
5.4. Les cibles	19
5.5. Utilisation d'iptables	20
5.6. Gestion la table NAT	21
6. Conclusion	23

Chapitre 3 : Installation et configuration de différents services réseaux

1. Introduction	24
2. Installation d'un serveur telnet	24
2.1. Le daemon inetd	24
2.2. TCP-Wrapper	24
2.3. Eléments de configuration	25
2.4. Les étapes d'installation d'un serveur telnet	25
3. Configuration serveur web apache	27
3.1. Configuration de base	27
3.2. Paramètres spécifiques à chaque serveur	27
3.3. Contrôle des accès à un répertoire	28
3.4. Les commandes d'utilisation d apache	29
4. Configuration d'un serveur de transfert des fichiers VsFTPd	31
4.1. Installation et configuration	31
4.2. Utilisation le client ftp FileZilla	33
5. Protocole SSH	33
5.1. Etapes d'installation et configuration d'un protocole SSH	34
6. Conclusion	35

Chapitre 4: Simulation d'un scénario de filtrage et sécurité sur un réseau virtuel

1. Introduction	36
2. La conception de notre réseau virtuel	36
3. Architecture complet de notre réseau virtuel avec la configuration des cartes réseaux	38
4. Les types de connexions réseaux avec Oracle VirtualBox	40
5. Les règles de configuration d'un pare-feu	41
6. Exemple de scénario de règle pour étudier le filtrage et la sécurité de notre réseau virtuel	42
6.1. Filtrage et sécurisation sur la table filter	42
6.2. Filtrage et sécurisation sur la table Nat	45
6.3. Autres règles testées sur le réseau virtuel (exemple réalisé).....	45
7. Conclusion	46
Conclusion générale	47

Introduction générale

De nos jours, la plus part des entreprises possèdent de nombreux postes informatiques qui sont en général reliés entre eux par un réseau local. Ce réseau permet d'échanger des données entre les divers collaborateurs internes à l'entreprise et ainsi de travailler en équipe sur des projets communs.

La possibilité de travail collaboratif apportée par un réseau local constitue un premier pas. L'étape suivante concerne le besoin d'ouverture du réseau local vers le monde extérieur, c'est à dire internet.

Pour parer à ces attaques, une architecture de réseau sécurisée est nécessaire. L'architecture devant être mise en place doit comporter un composant essentiel qui est le firewall. Cette 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.

Le firewall propose donc un véritable contrôle sur le trafic réseau de l'entreprise. Il permet d'analyser, de sécuriser et de gérer le trafic réseau, et ainsi d'utiliser le réseau de la façon pour laquelle il a été prévu. Tout ceci sans l'encombrer avec des activités inutiles, et d'empêcher une personne sans autorisation d'accéder à ce réseau de données.

L'objectif principale de notre projet est basé sur le teste et la configuré de notre pare feu sur un réseau composé par un ensemble de machines virtuelles. Les règles de teste sont appliquées sur un ensemble de services réseau comme le web, le service de connexion à distance telnet et le service de téléchargement des fichiers ftp.

Notre mémoire est décomposée en 4 chapitres :

Dans le chapitre un, nous présentons les notions de base de la virtualisation.

Le chapitre deux est consacré à l'étude des différents types et architecture d'un firewall.

Le troisième chapitre, nous présentons les étapes d'installation et configuration des services réseau comme apache, Telnet et ftp.

Dans le quatrième chapitre, nous présentons notre scénario de teste la sécurité et la protection de notre réseau virtuel selon les différents règles de firewall.

1. Introduction

Pour l'optimisation des coûts la virtualisation est devenue une réelle nécessité pour les entreprises. Introduction à un concept vieux de plus de 30 ans mais qui correspond à l'avenir des architectures informatiques

La virtualisation permet de faire fonctionner simultanément plusieurs systèmes d'exploitation sur une même machine physique. Dans l'esprit, il y aura plusieurs machines virtuelles sur une machine physique se partageant les ressources de celle-ci. L'évolution exponentielle des composants informatique et de leur puissance de calculs donnent de nouvelles optiques à l'informatique. La loi liée à l'évolution des composants informatique est la loi de Moore :

La loi de Moore veut que tous les 18 mois, une des 3 variables suivantes : la « puissance », la « vitesse » ou « l'espace » soit doublée.

Dans la logique, la virtualisation est basé sur le principe, au lieu de multiplier les machines physiques avec un seul système d'exploitation, on utilise une machine physique pour virtualiser plusieurs systèmes d'exploitations. La virtualisation a notamment été créée pour répondre à la problématique de la sous-utilisation des ressources matérielles.

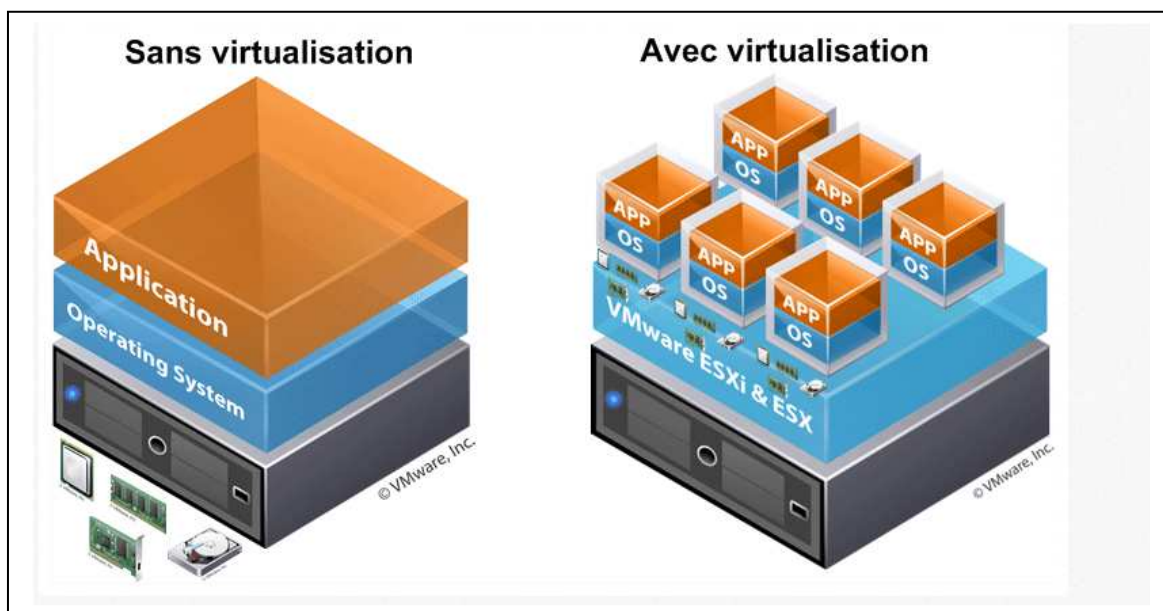


Figure 1.1 : virtualisation

2. Virtualisation

La virtualisation se compose:

D'une machine physique (Host)

Équipée de plusieurs cartes réseau, de beaucoup de mémoire et d'une grande capacité de stockage.

Un logiciel de gestions des ressources matérielles (Hyperviseur)

L'Hyperviseur permet à plusieurs systèmes d'exploitation de travailler et de partager les ressources d'une seule et même machine, le Host.

Des Machine virtuelle

Une machine virtuelle est un logiciel qui est installé sur un système d'exploitation appelé hôte (OS hôte, « host OS »). Cette machine virtuelle est capable d'émuler d'autres systèmes d'exploitation appelés invités (OS invités, voire OS clients, « guest OS »). Par abus de langage, on appellera une « machine virtuelle » (VM) le fichier représentant le système invité.

La machine virtuelle émule les périphériques (carte graphique, carte son, réseau, etc.) mais pas le processeur. A la différence des émulateurs de jeux de consoles, qui eux émulent également le processeur. Une machine virtuelle est donc plus performante qu'un émulateur « classique ».

3. Les types de virtualisation

3.1. Hyperviseur de type 1

L'hyperviseur de type 1 ou bare-metal est un outil qui s'interpose entre la couche matérielle et logicielle. Celui-ci a accès aux composants de la machine et possède son propre noyau. C'est donc par dessus ce noyau que les OS seront installés. Il pilote donc les OS à partir de la couche matérielle, il s'administre via une interface de gestion des machines virtuelles. Il est beaucoup plus puissant que les hyperviseurs de type 2 notamment grâce à sa proximité au matériel.

Les couches sont organisées comme suit :

- Couche matérielle
- Couche de virtualisation (hyperviseur)
- Virtualisation d'OS

Parmi ces hyperviseurs on trouve :

- VmWare vSphere
- Microsoft Hyper-V
- XEN
- KVM (open source)

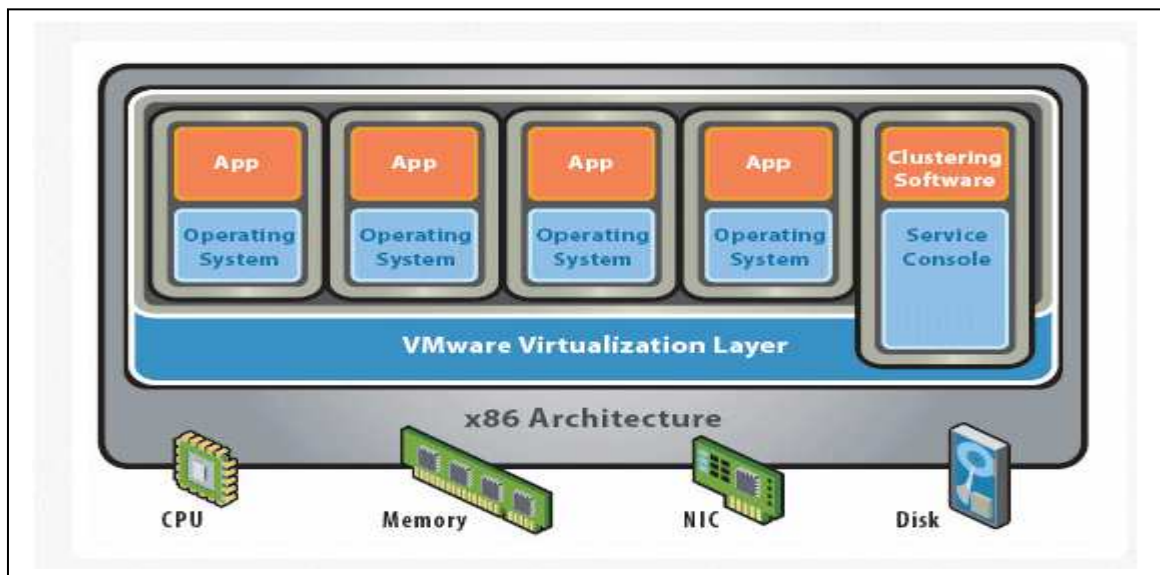


Figure 1.2 : Hyperviseur de type 1

3.2. Hyperviseur de type 2

L'hyperviseur de type 2 ou architecture hébergée est une application installée sur un système d'exploitation, elle est donc dépendante de celui-ci. Les performances sont réduites en comparaison des hyperviseurs de type car l'accès au matériel (CPU, RAM...) se fait via une couche intermédiaire. Néanmoins il propose une parfaite étanchéité entre les systèmes d'exploitations installés.

Les couches sont organisées comme suit :

- Couche matérielle
- Système d'exploitation hôte
- Couche de virtualisation
- Virtualisation d'OS

Parmi ces hyperviseurs on trouve :

- VmWare Workstation, Fusion, Player
- Oracle VirtualBox
- Microsoft Virtual PC
- QEMU (open source)

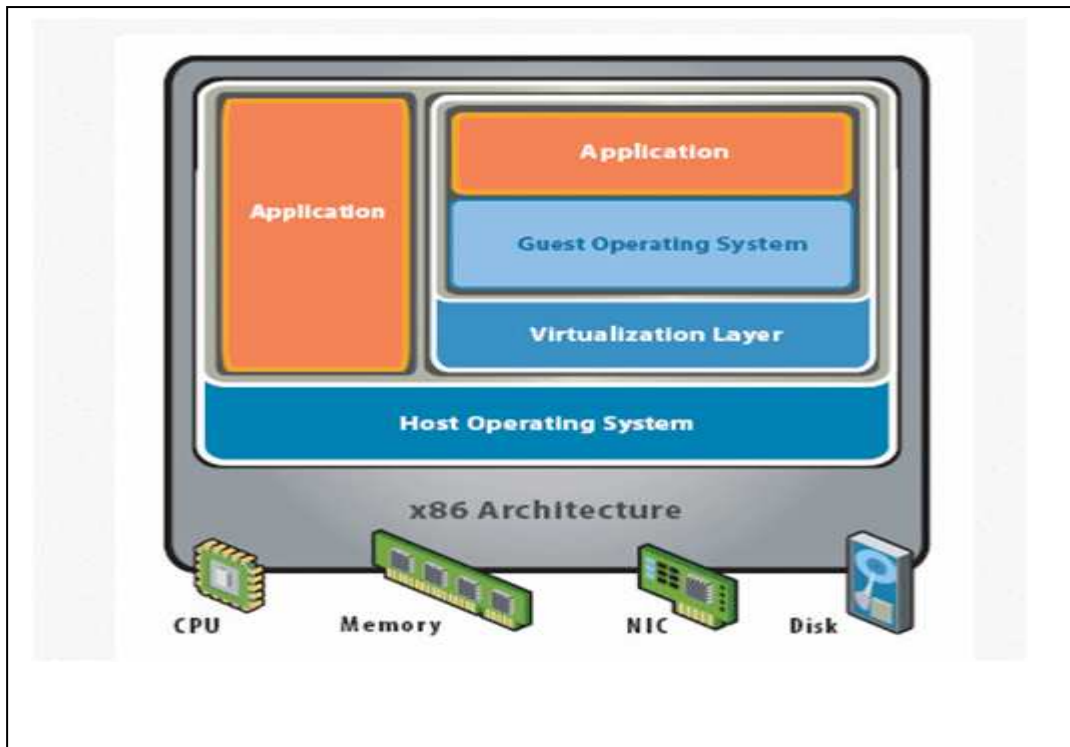


Figure 1.3 : Hyperviseur de type 2

4. Avantages de la virtualisation

Le premier avantage c'est de diminuer les coûts associés à l'achat d'équipement informatique. Dans une grosse entreprise les économies peuvent atteindre des centaines de milliers de dollars. Les plus petites peuvent quant à elles, envisager d'être plus productives avec moins d'équipement. Cela pourrait faire décoller des projets en attente faute de budget.

Réduction des coûts directs

- ❖ Réduction des frais associés à la désuétude informatique (Réduction du matériel)
- ❖ Réduction de la maintenance (Réduction du matériel)
- ❖ Coût d'électricité et de climatisation (coût significatif avec le nombre d'équipement)
- ❖ Meilleure compatibilité des logiciels avec les systèmes d'exploitation (Réponse à la désuétude avec les multi-OS, réponse aux incompatibilités avec la compartimentation)
- ❖ Aisance dans le processus de migration des plateformes (Multi-OS) (Intervention sur un nombre limité de machine et activation à distance).

Accroissement de la sécurité

- ❖ Meilleur contrôle grâce à une surveillance plus efficace des serveurs en temps réel.
- ❖ Portabilité et sécurité des activités en temps réel (Possibilité de redondance intersites en cas de sinistre). Avec la notion du « *cloud computing* », nous ajoutons encore plus de flexibilité.
- ❖ Meilleure sécurité grâce à la rapidité de se relever d'un désastre avec un plan de continuité.
- ❖

Accroissement de l'efficacité et de la productivité

- ❖ Meilleure efficacité informatique parce que vous faites plus avec moins (Moins de matériel et plus d'environnement compartimentés)
- ❖ Rapidité pour monter un nouveau serveur. En effet, tous les éléments sont toujours à portée de mains pour reprendre les activités.
- ❖ Meilleure protection grâce à la facilité de revenir en arrière en cas de sinistre
- ❖ Façon efficace de monter un environnement de test peu coûteux et complet. Avec une photo de l'environnement, il devient facile de tester sans tout bousiller et à sans dépenser trop en temps.
- ❖ Gestion des licences avantageuse motivant l'usage de la virtualisation, car vous aurez besoin de moins de licences pour être aussi efficace. Microsoft permet une gestion proactive de vos besoins en licences.

5. Inconvénients de la virtualisation

- C'est une technologie, non pas un protocole normalisé
- Mise en œuvre par des technologies différentes non standardisées..
- Repose sur des concepts différents, et Technologie « parfois » complexe à mettre en œuvre.
- Performances inégales selon la technologie de virtualisation employée
- Certaines technologies n'offrent pas de performances ou de stabilités suffisantes.
- Les serveurs n'ont plus d'E/S dédiées, chaque machine virtuelle partage les E/S sur disque.
- Baisse de performance possible à évaluer.
- Nécessité d'un serveur hôte plus puissant.
- Pertes plus importantes en cas de panne de la machine hôte plusieurs services indisponibles
- Nécessité de sauvegarder les machines virtuelles pour les relancer ailleurs en cas de problèmes

6. Virtualbox

Virtualbox est un logiciel de virtualisation des systèmes d'exploitation permettant de disposer de plusieurs systèmes d'exploitation sur une même machine en cours d'utilisation.

La virtualisation se faisant de plus en plus présente, VirtualBox trouve de plus en plus souvent sa place sur les postes simples. En effet, l'intérêt de virtualiser un système d'exploitation sur un serveur de production visible à partir de l'internet reste un cas rare.

Bien que VirtualBox soit nettement plus jeune que certains de ses concurrents comme par exemple VMWare, Sun a su rattraper son retard en proposant un logiciel tournant aussi bien sous Mac que sous Windows et également sous GNU/Linux (Linux, OpenBSD, FreeBSD).

Le virtualbox c'est l'objet de notre travail dans la configuration de notre pare feu.

7. Conclusion

La virtualisation est une technologie de plus en plus incontournable. Les environnements virtuels sont très en vogue au sein des entreprises de toutes tailles.

Il est vrai que les avantages de cette technologie sont nombreux en termes de productivité, de coûts et d'exploitation. En effet, elle permet des baisses de coûts importantes par la réduction du nombre de machines physiques, mais aussi par toutes les autres économies induites : énergie, temps de mise en œuvre.

La virtualisation d'un seul ordinateur physique n'est qu'un début. Durant, notre projet nous allons mettre en place une infrastructure virtuelle complète, en intégrant des machines virtuels interconnectés entre eux par un réseau virtuel.

1. Introduction

Nous allons étudier les firewalls (pare-feux ou mur de feux). Nous verrons ses fonctions, ses caractéristiques ainsi que ses limites. Nous étudierons également les différents types de configuration et architectures possibles.

Le firewall nous permet de résoudre certains problèmes liés à la sécurité, notamment lorsqu'on est connecté à internet.

Ce système a pour but de protéger une machine, un réseau domestique ou professionnel des attaques venant d'une autre machine distante ou de l'extérieur (souvent Internet), en gérant le trafic.

Suivant les types de firewall ou de leurs configurations, le " filtrage " peut être plus ou moins sévère.

2. Firewall outil de protection

Un pare-feu est un élément du réseau informatique, logiciel et/ou matériel, qui a pour fonction de sécuriser un réseau domestique ou professionnel en définissant les types de communication autorisés ou interdits.

Un réseau présent de nombreux risques s'il n'est pas correctement protégé contre les intrusions. Les risques sont variés, mais voici les plus courants et pouvant présenter un danger quelconque.

- ✓ espionnage industriel
- ✓ destruction de fichier
- ✓ vol d'informations

3. Technologies de filtrage

3.1. Le filtrage de paquet

Un Firewall à filtrage de paquets est une technologie de première génération qui analyse le trafic sur la couche transport. Chaque paquet IP est examiné de façon à voir s'il correspond à une des règles définissant les types de flux permis. Ces règles permettent de savoir si une communication est autorisée, en se basant sur l'entête de la trame IP et sa direction (interne vers externe et vice versa).

Le filtrage de paquets permet de manipuler) le transfert de données, en se basant sur les contrôles suivants : réseau physique sur lequel le paquet arrive

- L'adresse d'où le paquet est supposé arriver (adresse source)
- L'adresse destination
- Le type de protocole de transport utilisé (TCP, UDP, ICMP)
- Les ports source / destination (pour TCP ou UDP)

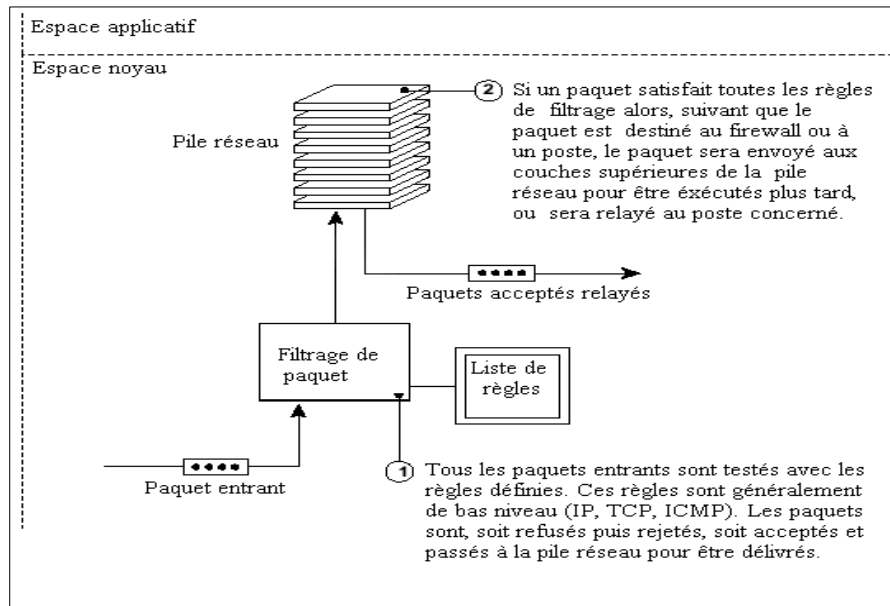


Figure 2.1 : Le filtrage de paquet

L'inspection complète d'un paquet suit l'algorithme suivant :

- Si aucune règle de permission n'est trouvée, le paquet est rejeté
- Si on trouve une règle qui autorise cette communication, on autorise la connexion
- Si on trouve une règle qui interdit la communication, le paquet est rejeté

Comme ce type de firewall n'inspecte pas la couche application du paquet, et ne supervise pas l'état des connexions, c'est la solution la moins sûre des technologies de firewall. Elle laisse passer les paquets avec un minimum d'éléments de décision. En contrepartie, comme il exécute moins de fonctions que les firewalls de technologie différente, c'est le moyen le plus rapide et il est souvent implémenté dans des solutions hardware telles que les routeurs IP.

Pour conclure voici les avantages du filtrage de paquets :

- Les filtres de paquets sont généralement plus rapides que les autres technologies, car ils procèdent à moins d'opérations. Ils sont aussi plus faciles à implémenter en dur.
- Une règle suffit à bannir une source spécifiée.
- Cette technologie est totalement transparente pour l'utilisateur.
- En association avec NAT, on peut protéger les adresses IP internes des utilisateurs externes.

Et ses inconvénients

- Les filtres de paquets ne comprennent pas la couche application. Ils ne peuvent donc pas assurer la sécurité pour des services basiques tels que PUT ou GET dans FTP. Pour cette raison, c'est la moins fiable des solutions.
- Ils ne gardent pas d'informations sur les différentes connexions.
- Ils ne peuvent pas manipuler les informations contenues dans un paquet.

3.2. Firewalls de niveau circuit

Un firewall de niveau circuit est une technologie de seconde génération qui accepte le fait qu'un paquet soit une demande de connexion ou qu'il appartienne à une connexion ou un circuit virtuel existant.

Les paquets de données ne sont transmis que lorsque la connexion est établie. Le firewall maintient une table des connexions valides et ne laisse passer les paquets que s'ils correspondent à une de ces connexions. A la déconnexion, la ligne correspondante est effacée.

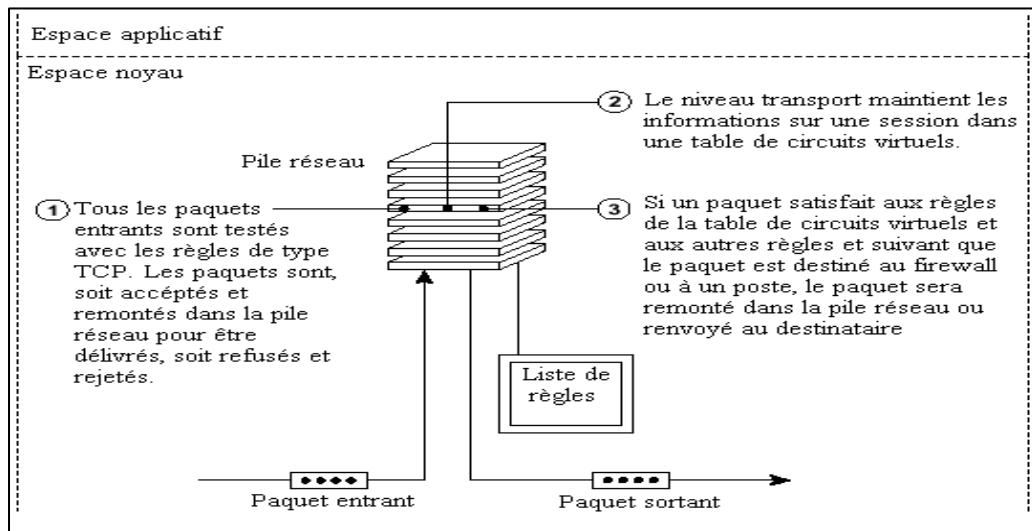


Figure 2.2 : Firewalls de niveau circuit

Quand une connexion est établie, le firewall stocke les informations suivantes.

- Un numéro unique pour identifier la connexion.
- L'état de la connexion : *handshake*, *established* ou *closing*.
- Les informations de séquence.
- L'adresse IP source (d'où les données arrivent).
- L'adresse IP destination (où les données vont).
- L'interface physique d'entrée.
- L'interface physique de sortie.

En utilisant ces informations, le firewall vérifie l'entête de chaque paquet pour déterminer si la machine source a la permission d'envoyer des données à la machine cible, et si celle-ci a la permission de les recevoir.

Ces firewalls n'ont qu'une compréhension limitée des protocoles utilisés dans la couche réseau. Ils ne peuvent détecter qu'un protocole de niveau transport, comme TCP. Tout comme le filtre de paquets, le firewall de niveau transport applique une liste de règles maintenues dans le noyau TCP/IP.

Souvent, le firewall ré adresse le paquet pour qu'il semble provenir du firewall plutôt que de l'hôte interne. En conservant les informations sur chaque session, il peut facilement faire correspondre les réponses externes à l'hôte interne approprié.

Pour résumer, voici les avantages du firewall de niveau circuit :

- Il est généralement plus rapide que les firewalls de couche application, car il procède à moins de vérifications.
- Il peut protéger un réseau en interdisant les connexions entre certaines sources Internet et les ordinateurs internes.
- En association avec NAT, on peut protéger les adresses IP internes des utilisateurs externes.

Et ses inconvénients :

- Il ne peut restreindre les accès que pour le protocole TCP.
- Il ne peut contrôler les protocoles de niveau supérieur.
- Il n'offre pas de services supplémentaires tels que le cache HTTP, le filtrage d'URL, et l'authentification car ils ne comprennent pas les protocoles mis en jeu.
- Il est difficile de tester l'efficacité des règles « accept » et « deny »

3.3. Firewall de couche application

Un Firewall de couche application est un firewall de troisième génération qui vérifie la validité des données au niveau application avant d'autoriser une connexion. Il examine le paquet et maintient l'état et l'historique de la connexion. En plus, ce type de firewall permet d'utiliser une authentification.

Pour résumer, voici les avantages du firewall de niveau application :

- Les services *proxy* comprennent et renforcent les protocoles de haut niveau comme FTP ou HTTP.
- Ils tiennent à jour les informations concernant les données transitant. Ils fournissent des informations sur les états des communications, ainsi que sur les sessions.
- Ils peuvent interdire les accès à certains services et en autoriser d'autres.
- Ils peuvent traiter les informations d'un paquet, et les modifier.
- Ils n'autorisent pas les communications directes.
- Ils donnent aux utilisateurs l'impression de communiquer directement avec l'extérieur.
- Ils mettent à disposition quelques services supplémentaires comme le cache http, le filtrage d'URL ou une authentification d'utilisateur.
- Ils sont un moyen pratique pour générer des rapports d'audit, et permettent à l'administrateur de superviser les tentatives de violation de la sécurité.
- Les applications du serveur *proxy* ne s'exécutent pas forcément tous sur la même machine, le proxy peut très bien router certains services (comme HTTP ou FTP) sur une autre machine serveur, de façon à diminuer sa charge de travail.

Et ses inconvénients

- Sa mise en place, demande de remplacer l'ancienne pile réseau dans le firewall.
- Comme il écoute sur les mêmes ports que les serveurs, on ne peut pas faire tourner de serveur sur un firewall.
- En général, un nouveau service *proxy* doit être écrit pour chaque protocole. Approximativement, il faut considérer un délai de 6 mois entre l'arrivée d'un nouveau protocole et celle du *proxy* qui lui sera propre.
- Enfin c'est toujours une charge supérieure pour les utilisateurs qui doivent souvent configurer leurs clients ou encore taper des mots de passe pour pouvoir faire fonctionner leur client.

3.4. Filtrage dynamique de paquets

Un firewall à filtrage de paquet est un firewall de quatrième génération qui permet la modification des règles de filtrage à la volée. Ce type de filtrage est utilisé pour limiter les accès par UDP au réseau.

Le filtrage dynamique de paquet possède les mêmes avantages et inconvénients que le firewall de première génération, à la différence près qu'il n'autorise pas les paquets UDP non sollicités, à l'intérieur du réseau.

Ce filtrage est pratique pour rendre possible le franchissement de votre périmètre de sécurité à votre serveur DNS interne (Domain Name Server) qui fonctionne souvent en mode UDP. Le serveur interne envoie une requête à un DNS extérieur pour un hôte inconnu, et la réponse pourra franchir le firewall sans problèmes.

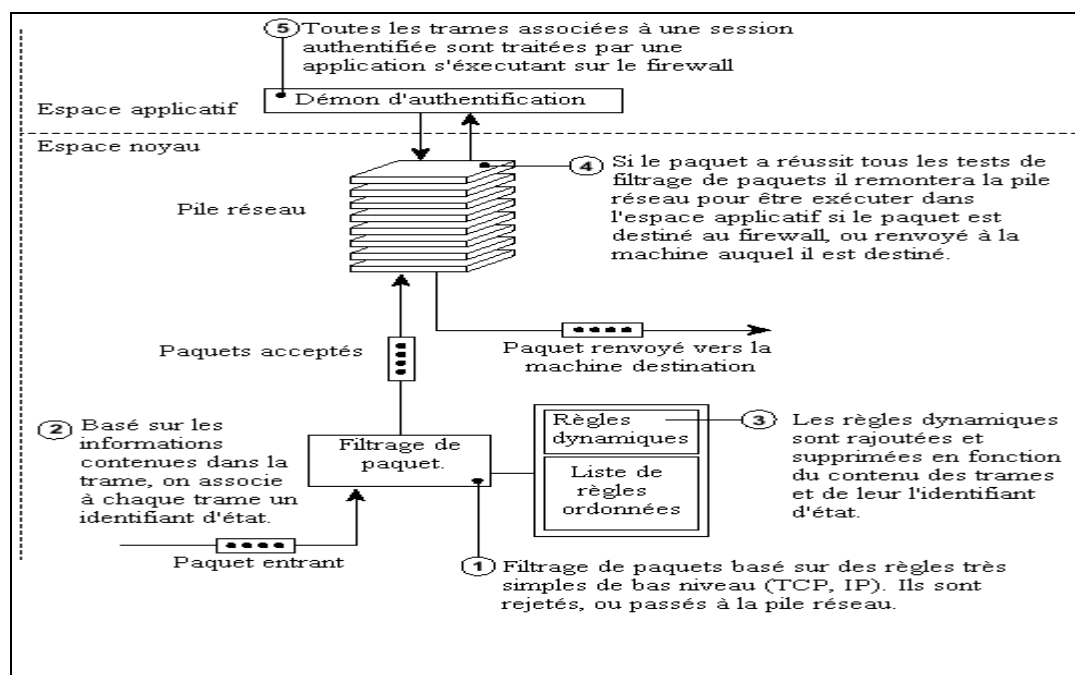


Figure 2.3 : Filtrage dynamique de paquets

4. Architectures firewall

Le firewall n'est pas seulement une solution logicielle de sécurité implantée sur une machine, c'est aussi une architecture réseau de machines filtrantes.

L'approche simpliste d'un firewall localisé sur une machine jouant le rôle de grand chef d'orchestre n'a plus cours à présent dans les grandes entreprises, car elle représente un goulet d'étranglement pour le débit Internet et elle est trop peu sécurisée en cas de panne ou faille dans cette unique défense.

La mise en place de plusieurs filtres de différents niveaux assurent une meilleure étanchéité du réseau, et un meilleur débit, mais ils s'accompagnent d'un coût plus élevé.

4.1. Firewall avec routeur de filtrage

La solution Firewall la plus simple, mais aussi la moins sûre, se borne au réseau. On l'obtient en configurant le routeur qui assure la connexion avec l'Internet. L'image suivante illustre cette solution appelée Firewall avec routeur de filtrage.



Figure 2.4 : Firewall avec routeur de filtrage

Cette solution permet de réaliser les différents serveurs d'un Intranet sur plusieurs systèmes. Le routeur de filtrage contient les autorisations d'accès basées exclusivement sur les adresses IP et les numéros de port.

Les filtres mis en œuvre dans le routeur pourront être (entre autres) :

- "IP source-routing" invalidé
- "IP-Spoofing" (mascarade IP) interdite
- "sites louches" filtrés en entrée
- Applicatifs comme : X11, RSH, R-Command (port-mapper), Finger, TFTP, etc.

On peut aussi envisager une solution identique à la précédente pour la partie filtrage, mais avec une station de surveillance du trafic après le routeur.

Avantage : facilité de configuration, bon marché, de plus il fournit des traces exploitables, et surtout la possibilité d'alarmes pour :

- ✓ une vérification du bon fonctionnement des filtres du routeur.
- ✓ il y ait encore un peu de temps pour réagir si le routeur est compromis.

Inconvénient : lorsque le routeur est contourné ou paralysé, le réseau entier est ouvert

4.2. Passerelle double ou réseau bastion



Figure 2.5 : Passerelle double ou réseau bastion

La passerelle double est la possibilité la plus simple pour réaliser un Firewall d'application n'autorisant aucun trafic IP entre les réseaux.

La machine «coupe-feu» cumule les fonctions de filtrage, de PROXY, de passerelle applicative et de trace.

Elle :

- Filtre tout trafic entre le monde extérieur et le réseau local.
- Joue le rôle de serveur (store and forward) pour NNTP (news), SMTP (mail), et de proxy pour HTTP.
- Sert de passerelle avec authentification pour les applicatifs telnet, rlogin, FTP.
- rend les services équivalents à la machine de LOG de la solution précédente

Pour simplifier le rôle du Proxy/Filtre, nous pourrions faire un filtre sélectif entre le monde extérieur et le réseau local et permettre ainsi un trafic direct pour certains services comme SMTP, NNTP, HTTP, mais cela demanderait un logiciel de routage spécifique.

Avantage : bon marché.

Inconvénient :

- Il ne faut pas que cette machine présente de faiblesses, car c'est le seul rempart contre l'adversité !
- Du fait de tout ce qu'elle doit faire (routage et application), une telle configuration pourrait rencontrer des problèmes de performance.

4.3. Firewall avec réseau de filtrage

La combinaison des deux méthodes est ici plus sûre et efficace. En ce qui concerne le réseau, un routeur sous écran est configuré de façon à n'autoriser les accès de l'extérieur et de l'intérieur que par l'intermédiaire du réseau bastion sur lequel fonctionnent tous les serveurs assurant les serveurs Internet. Cette possibilité est appelée Firewall avec réseau de filtrage. L'image suivante illustre cette solution

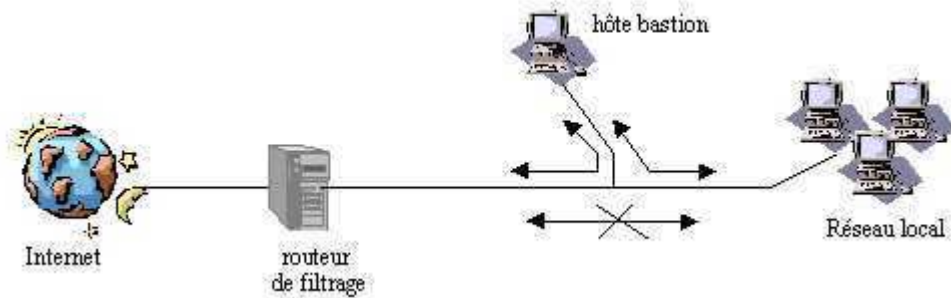


Figure 2.6 : Firewall avec réseau de filtrage

Firewall avec réseau de filtrage dans lequel seuls les accès au réseau bastion sont autorisés. Pour la grande majorité des entreprises, cette solution est sûre et abordable, car les prestataires Internet assurent la seconde partie de la protection à l'autre bout de la ligne. En effet, votre entreprise y est également connectée à un routeur, et le trafic de données est réglé par un serveur Proxy en ce qui concerne la couche application. Les pirates doivent par conséquent franchir deux obstacles.

Avantage : bon marché et sûr lorsque le prestataire est équipé en conséquence.

Inconvénient : le coût d'investissement est plus élevé.

4.4. firewall avec sous-réseau de filtrage

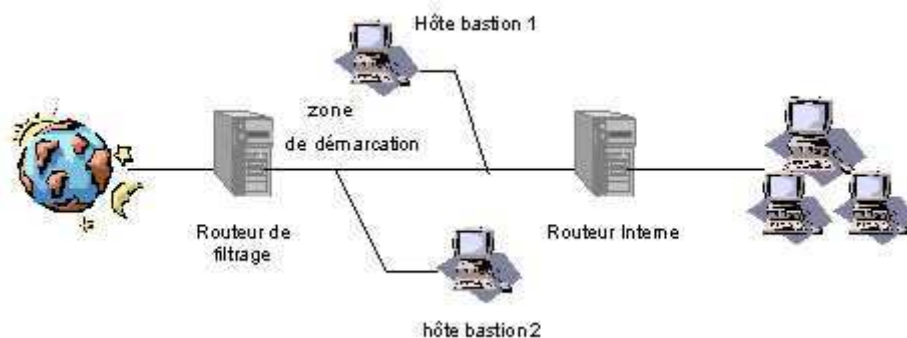


Figure 2.7 : Firewall avec sous-réseau de filtrage

Cette solution est de loin la plus sûre, mais également la plus onéreuse. Un Firewall avec sous-réseau de filtrage se compose de deux routeurs sous écran. L'un est connecté à Internet, et l'autre à l'Intranet/LAN. Plusieurs réseaux bastions peuvent s'intercaler pour former entre ces deux routeurs, en quelque sorte, leur propre réseau constituant une zone tampon entre un Intranet et l'Internet appelée «zone démilitarisée».

De l'extérieur, seul l'accès aux réseaux bastions est autorisé. Le trafic IP n'est pas directement transmis au réseau interne. De même, seuls les réseaux bastions, sur lesquels des serveurs Proxy doivent être en service pour permettre l'accès à différents services Internet, sont accessibles à partir du réseau interne.

Pour s'introduire sur le réseau d'entreprise à travers ce Firewall, il faut franchir les deux routeurs, ainsi que les réseaux bastions intercalés

Le Routeur interne :

- Autorise le trafic entre le bastion 1 et les machines internes et inversement.
- Interdit tout autre trafic.

Le Routeur externe :

- Filtre le trafic entre le monde extérieur et le bastion 2.
- Interdit tout autre trafic direct(donc pas de trafic direct entre le réseau interne et l'extérieur).

Les deux bastions peuvent discuter sans aucune règle => zone démilitarisée (DMZ)

Le bastion interne :

- Assure les fonctions de DNS vis à vis du réseau interne en envoyant ses requêtes au bastion externe.
- Assure les fonctions de *proxy* avec authentification pour les applications distantes (Telnet, Rlogin, FTP)
- Assure le relais du Mail sortant(SMTP).

Le bastion externe :

- Filtre au niveau applicatif les paquets en direction du réseau interne
- Assure le relais du Mail entrant(POP).
- Assure les fonctions de DNS vis à vis du réseau externe.

Avantage : système Firewall très sûr.

Inconvénients : coût d'investissement élevé, et il faut fournir un effort d'installation, et d'administration important.

5. Fonctionnement du pare-feu sous Linux : NetFilter/Iptables

NetFilter est actuellement le filtrage le plus utilisé sous Linux. Il est disponible depuis la version 2.4 du noyau et remplace donc ipchains présent dans la version 2.2. NetFilter est composé de 2 parties :

d'une part, NetFilter proprement dit qui doit être compilé dans le noyau (« en dur » ou sous forme de module), d'autre part la commande iptables.

5.1. 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. Dans les paragraphes suivant les principaux types de tables, de chaînes et cibles seront détaillées.

5.2. Les tables

Il existe par défaut dans NetFilter une seule table, la table filter. Cette table permet de filtrer les paquets entrant, sortant et transitant avec respectivement les chaînes INPUT, OUPUT et FORWARD.

Grâce à l'ajout du module iptable_nat, une nouvelle table est accessible, la table nat. Comme son nom l'indique, elle contient les chaînes qui vont s'appliquer pour la translation d'adresses mais aussi de port. Les chaînes disponibles avec ce module sont : PREROUTING, POSTROUTING, OUTPUT.

On dispose également de nouvelles cibles notamment MASQUERADE, DNAT, SNAT. Ces deux dernières cible sont respectivement utilisées pour modifier l'adresse destination et l'adresse source des paquets.

Une troisième table disponible est la table MANGLE. Cette table est utilisé notamment lors de la mise en place de la QoS pour marquer les paquets.

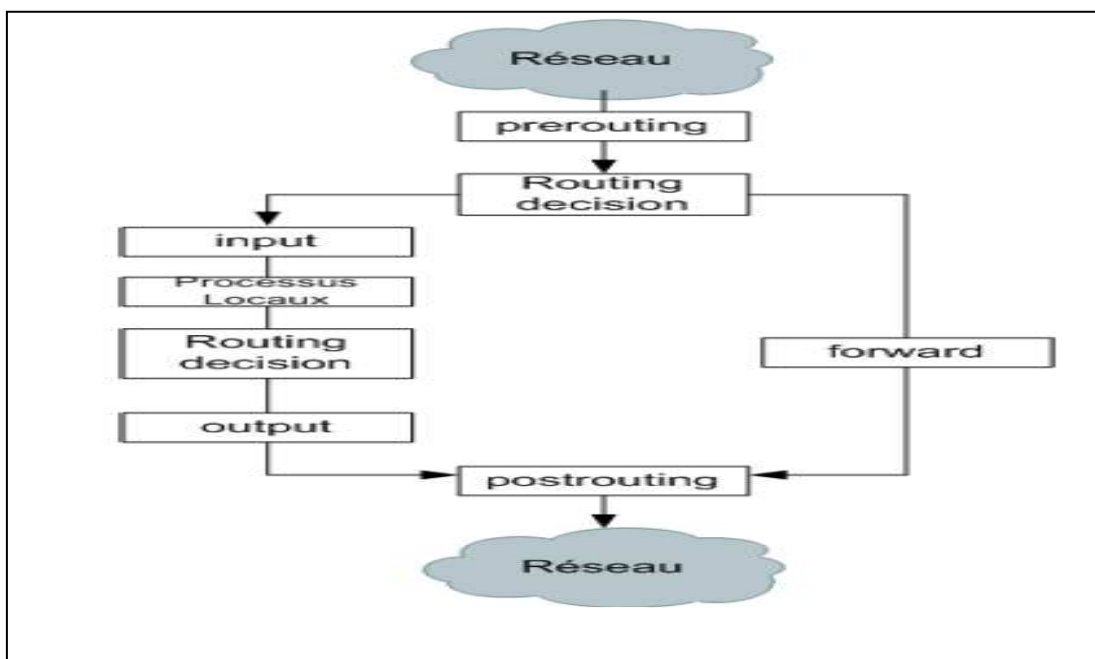


Figure 2.8 : fonctionnement d'un firewall

5.3. Les chaînes

INPUT : Cette chaîne est utilisée pour les paquets étant à destination des applications du firewall. A ce stade, les paquets sont prêts à être envoyé aux applications.

OUTPUT : Cette chaîne est utilisée pour les paquets sortant des applications du firewall. A ce stade, les paquets ont donc déjà été traités, ou générés par les applications.

FORWARD : Cette chaîne filtre les paquets passant d'une interface à une autre du firewall, c'est à dire qu'ils ne sont pas destinés à une application présente sur le firewall. Ces paquets ne passent pas par les chaînes INPUT et OUTPUT et ne passent jamais par la couche applicative. Dans ce cas, le firewall se comportera comme une passerelle.

PREROUTING : Quand les paquets arrivent au niveau du firewall, ils sont dans un état non modifié. C'est à dire qu'il n'y a encore eu aucun traitement quel qu'il soit sur celui-ci au niveau du firewall.

Cette chaîne est utilisée afin de faire des traitements particuliers sur les paquets en arrivé avant d'effectuer leur filtrage à proprement dit. Il est utilisé, par exemple, dans les cas d'utilisation de destination NAT ou DNAT, qui correspond à la modification de l'adresse IP destination.

POSTROUTING : Quand les paquets sont prêts à être envoyés sur l'interface réseau. Ils ont donc été traités par les applications, et router par le firewall.

Tous les traitements sur ces paquets sont alors terminés. Il est utilisé, par exemple, dans le cas de source NAT ou SNAT, qui correspond à la modification de l'adresse IP source (utile pour accéder au réseau Internet avec une adresse IP privé).

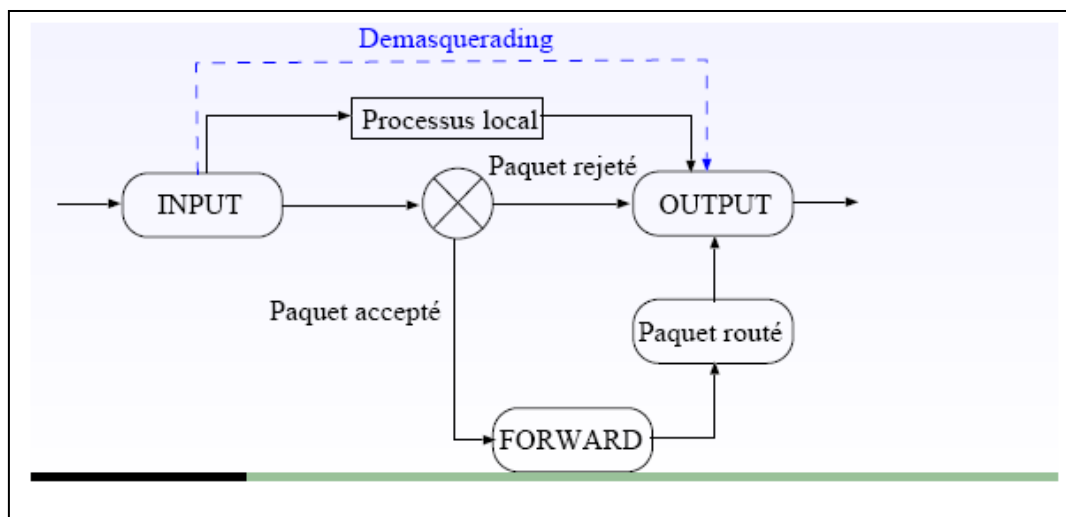


Figure 2.9 : iptables

Voici comment nous pouvons résumer l'utilisation des chaînes dans un firewall. Nous constatons bien que les chaînes INPUT et OUTPUT sont à destination ou départ du noyau Linux du firewall.

Cela valide le fait qu'elles ne sont utilisées que pour les services que firewall lui-même. Nous constatons de même que la chaîne FORWARD ne passe jamais par le noyau du firewall, ces paquets ne sont donc pas traités par les processus externe au firewall.

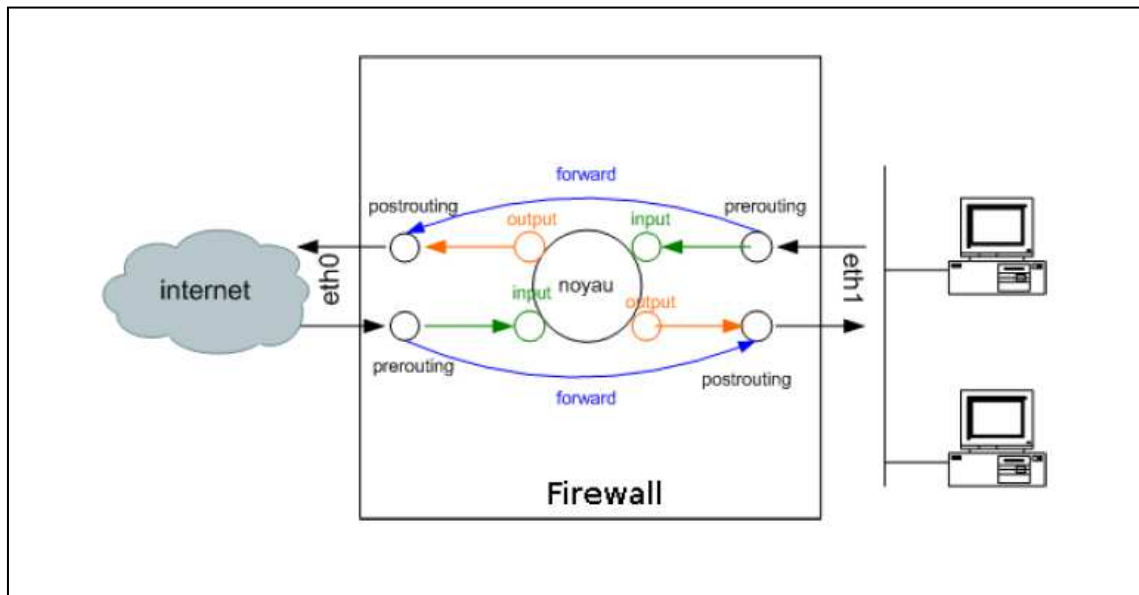


Figure 2.10 : firewall

5.4. Les cibles

Un firewall est donc une suite de règles qui spécifient des critères. Si un paquet ne correspond pas à une règle c'est la prochaine règle de la chaîne qui est utilisée, si il correspond la règle va « sauter » (jump) vers une autre règle (target, cible).

Cette cible peut être une autre règle définie par la personne en charge de la configuration du firewall, mais, le plus souvent, ce sont des cibles particulières, définies par Iptables qui sont utilisées.

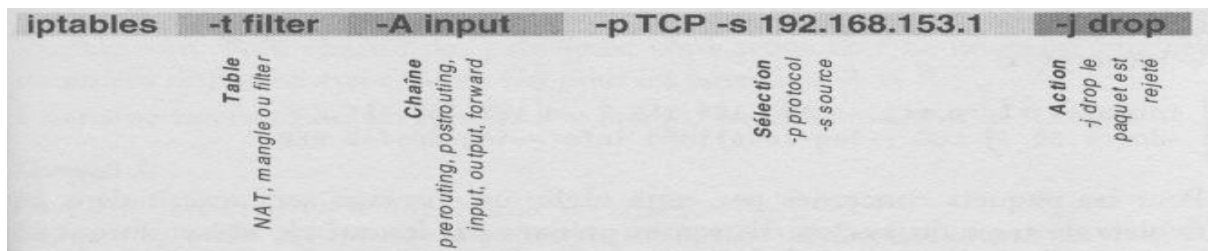
Parmi les cibles définies par Iptables trois sont fréquemment utilisées : ACCEPT, DROP et REJECT. Ces règles sont dites terminales car elles ne pourront pas être utilisées pour effectuer un saut vers une autre règle.

- **ACCEPT** signifie qu'on laisse passer le paquet à travers le firewall.
- **DROP** signifie que le paquet est purement et simplement jeté. L'hôte source du paquet ne sera pas prévenu, le cas est identique à la perte du paquet.

- **REJECT** signifie que le paquet est rejeté. A la différence de DROP, un paquet d'erreur est transmis à l'émetteur du paquet rejeté. Ainsi celui-ci est prévenu que le paquet a été rejeté et pour donc agir en conséquence.

5.5. Utilisation d'iptables.

Syntaxe général pour écrire une règle IPTables.



1. Par défaut il y a trois chaînes INPUT, OUTPUT et FORWARD que tu ne peux pas effacer. Regardons les opérations pour administrer les chaînes :
2. Créer une nouvelle chaîne (-N).
3. Effacer une chaîne vide (-X).
4. Changer la règle par défaut pour une chaîne de départ (-P).
5. Lister les règles dans une chaîne (-L).
6. Retirer les règles d'une chaîne (-F).
7. Mettre à zero les compteurs de bits et de paquets d'une chaîne (-Z).

Il y a plusieurs manières de manipuler une règles dans une chaîne :

1. Ajouter une nouvelle règle à la chaîne (-A).
2. Insérer une nouvelle règle à une position dans la chaîne (-I).
3. Remplacer une règle à une position dans la chaîne (-R).
4. Supprimer une règle à une position dans la chaîne (-D).
5. Supprimer la première règle qui convient dans une chaîne (-D).

Liste des actions possibles

-j ACCEPT	Le paquet est accepté.
-j DROP	Le paquet est rejeté.
-j REJECT	Le paquet est rejeté, l'expéditeur est averti de l'indisponibilité du service.
-j QUEUE	Le paquet est envoyé à une application.
-j LOG	Le paquet est envoyé au système « syslog ».
-j MARK	Le paquet est marqué.
-j TOS	Modifie le « Type Of Service » du paquet.
-j MIRROR	Renvoie le paquet à l'expéditeur.
-j SNAT	L'adresse source du paquet est translatée.
-j DNAT	L'adresse destination du paquet est translatée.
-j MASQUERADE	L'adresse source du paquet est translatée.
-j REDIRECT	R-direction d'un port vers un autre.

5.6. Gestion la table NAT

Configurer la table NAT du pare-feu

La table nat permet de faire la translation d'adresse réseau (NAT) sur différents paquets. La traduction d'adresse est gérée avec les chaînes PREROUTING, POSTROUTING et OUTPUT.

Voici le structure de test utilisé pour les exemples suivants :

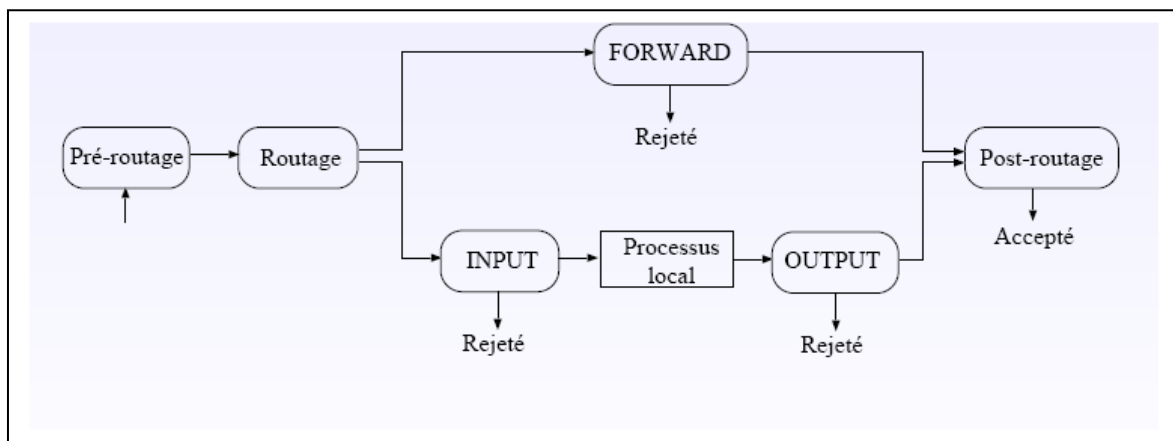
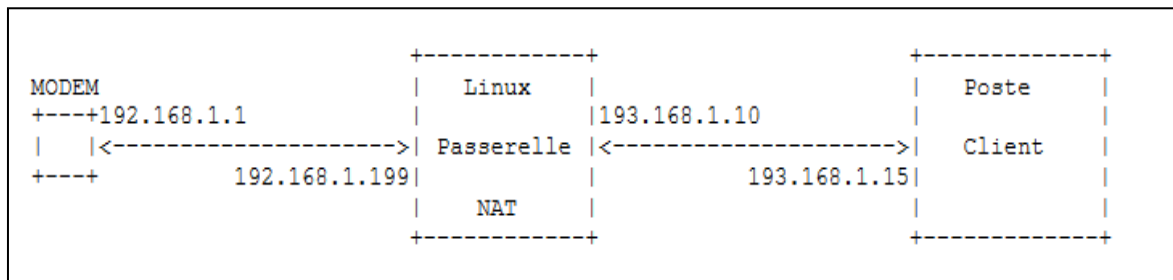


Figure 2.11 : table NAT

Le DNAT ou NAT Destination

La chaîne PREROUTING (avant routage) permet de modifier que l'adresse de destination mais conserve l'adresse source. C'est ce qu'on appel faire du DNAT (NAT destination).

Dans cet exemple, on va rediriger le trafic en destination du réseau "195.111.222.0" vers le réseau "193.168.1.0".

```
iptables -t nat -A PREROUTING -d 195.111.222.0/24 -j DNAT --to-destination 193.168.1.0/24
```

Le SNAT ou NAT Source

La chaîne POSTROUTING (après routage) permet de modifier que l'adresse source mais conserve l'adresse de destination. C'est ce qu'on appel faire du SNAT (NAT source).

Dans cet exemple, on va substituer l'adresse source du trafic privé sortant vers l'extérieur par une des trois adresses publique.

```
iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j SNAT --to-source 192.168.1.5-192.168.1.8
```

L'IP Masquerade

Le principe est d'utiliser une adresse IP publique source pour cacher derrière elle toutes les adresses IP du réseau privé.

Dans cette exemple avec du SNAT, tous les paquets provenant du réseau privé 193.168.1.0 sera perçu comme provenant de l'IP public 192.168.1.199.

```
iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j SNAT --to-source 192.168.1.199
```

Il y a aussi l'option "MASQUERADE" qui a le même but.

```
iptables -t nat -A POSTROUTING -s 193.168.1.0/24 -j MASQUERADE
```

Vous pouvez utiliser d'autres options, comme choisir par rapport a votre interface réseau , ici "eth0" et au numéro de port, dans ce cas "80".

```
iptables -t nat -A POSTROUTING -o eth0 --dport 80 -j MASQUERADE
```

Afficher la table NAT

Pour afficher la table de NAT du pare-feu.

```
# iptables -t nat -L
```

Supprimer une règle de la table NAT(-D)

Pour commencer, on va lister les règles avec un numéro par ligne.

```
# iptables -t nat -L --line-numbers
Chain PREROUTING (policy ACCEPT)
num target prot opt source destination

Chain POSTROUTING (policy ACCEPT)
num target prot opt source destination
1 MASQUERADE all -- 193.168.1.0/24 anywhere

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
```

Dans notre exemple on va supprimer la règle "1" de la chaîne "POSTROUTING".
Syntaxe:

```
iptables -t nat -D [chaîne] [numéro_de_ligne]
```

```
# iptables -t nat -D POSTROUTING 1
```

Exemple de script

Le script ci-dessous est un exemple de script permettant la configuration de NetFilter à l'aide de la commande iptables. Ce type de script doit être placé à l'emplacement adéquat pour qu'il soit appelé dès que les interfaces réseaux sont activées. Par exemple sous ubuntu il doit se trouver dans le dossier /etc/network/.

L'exemple fournis ici peut être appliqué à une machine personnelle connecté à Internet via un modem.

```
#!/bin/sh

#ppp0: internet

#Suppression des règles prédéfinies pour toutes les tables :
iptables -F
iptables -t nat -F
iptables -t mangle -F

#Suppression de toutes les règles de l'utilisateur :
iptables -X

#Politique par défaut (tout rejeter) :
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# l'interface loopback du firewall peut émettre dans tous les sens :
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# les connections invalides sont refusées :
iptables -A INPUT -m state --state INVALID -j DROP
iptables -A FORWARD -m state --state INVALID -j DROP

# les connections établies ou assimilables sont acceptées en entrées :
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Le firewall peut émettre comme il veut sur ppp0 :
iptables -A OUTPUT -o ppp0 -j ACCEPT
```

6 .conclusion

Comme on peut le constater, les firewalls possèdent de multiples capacités qui peuvent différer en fonction de leurs types. Cette multitude de solutions impose donc une étude rigoureuse de la sécurité devant être mise en place. En effet, le système informatique d'une centrale nucléaire n'aura pas le même besoin en termes de sécurité qu'un particulier et aura donc par conséquent des équipements différents.

Il est également nécessaire de préciser que le firewall est seulement un composant de sécurité, il ne protégera donc pas à lui seul un réseau. Il est nécessaire de l'inclure dans une démarche qui prendra en compte d'autres paramètres tel que la mise à jour des applications.

Dans le chapitre suivant, nous présentons l'installation et la configuration des différents services réseau.

1. Introduction

Dans ce chapitre, nous décrivons les étapes d'installation et configuration des différents services réseau comme le service de connexion à distance telnet, web apache et vsftpd de transfert des fichiers. Les différents services installés seront utilisés pour tester notre firewall.

2. Installation d'un Serveur Telnet

Telnet est un protocole qui permet l'émulation de terminal à distance sur un serveur (connexion à distance à un serveur en utilisant le port 23 sous Unix/Linux).

2.1. Le daemon inetd

Toute application fonctionnant sous TCP/IP est basée sur le modèle client/serveur. Par exemple quelqu'un se connectant via telnet à un hôte distant « active » chez l'hôte le service serveur telnetd.

Chaque serveur est sur une machine en attente d'une connexion sur un port particulier. Dans les premières versions d'Unix-TCP/IP chaque application telnet avait son propre serveur qui était lancé au démarrage de chaque machine comme un "daemon". Cette stratégie encombrait inutilement la table des processus (autant de serveurs que de services). Ces services sont dits fonctionnant en mode « autonome » ou « standalone ».

Le daemon INETD est un « super » serveur, à l'écoute sur plusieurs ports et qui se charge de recevoir les demandes de connexion de plusieurs clients telnet et de lancer le serveur correspondant à la demande. A son démarrage il consulte les fichiers:

- /etc/services qui contient la liste générale des services TCP/IP avec leur numéro de port et le protocole de transport associé.

- /etc/inetd.conf qui contient la liste des services activés sur une machine donnée

Extrait de /etc/services :

```
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp mail
pop3 110/tcp # Post Office
```

2.2. TCP-Wrapper

TCP-Wrapper est un outil de sécurité réseau qui permet de contrôler les accès, les tentatives de connexion sur une machine donnée par journalisation syslogd. Si inetd reçoit une demande de connexion sur le port 23 il va lancer telnetd.

Tcp-wrapper sert d'enveloppe. Il vient « s'intercaler » entre le daemon inetd et le serveur à démarrer. Quand une demande de service TCP/IP (en réalité TCP ou UDP) arrive sur un port donné, inetd va lancer **tcpd** (daemon correspondant à Tcpwrapper) au lieu d'activer directement le service demandé (telnetd, ftpd, pop3...).

Tcpd prend en charge la requête et met en place ses mécanismes de contrôle

2.3. Eléments de configuration

/etc/inetd.conf qui contient la liste des services activés sur une machine donnée

Extrait de /etc/inetd.conf

```
ftp      stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftpd
telnet   stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
```

Exemple:

```
# Fichier /etc/hosts.deny          # Fichier /etc/hosts.allow
# interdit tous les accès ftp      # autorise les accès ftp
à la machine in.ftpd:ALL          venant de cli1 in.ftpd :cli1.master.dz
```

2.4. Les étapes d'installation d'un serveur Telnet

Etape 1 : installation d'un client telnet (par défaut est installé sous linux ubuntu)

Etape 2 : installation d'un serveur telnetd (paquet : telnetd_0.17-36_i386.deb)

Etape3 : installation d'un super-serveur internet (paquet : openbsd-inetd_0.2...deb)

Etape 4 : installation d'un service de sécurité tcpd (paquet tcpd.deb)

Etape 5 : vérifier si le fichier de configuration de super-serveur openbsd-inetd contient la ligne (/etc/inetd.conf) suivante :

```
telnet stream  tcp      nowait  root    /usr/sbin/ftpd  usr/sbin/in.telnetd
```

etape 6 : redémarrer le super-serveur par la commande: /etc/init.d/openbsd-inetd restart

Etape 7 : vérification si le service est en état marche : exécuter la commande suivante

Ps aux | grep telnet

Etape 8 : teste si le service telnet marche localement : exécuter la commande suivante

telnet localhost ou telnet 127.0.0.1

```
debian2:~# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Debian GNU/Linux 4.0
debian2 login:

debian2:~# netstat -natupw | grep ESTABLISHED
tcp        0      0 127.0.0.1:47876      127.0.0.1:23        ESTABLISHED2283/telnet // port serveu
tcp        0      0 127.0.0.1:23        127.0.0.1:47876    ESTABLISHED2284/in.telnetd: lo // port client
```


etape 9 : créer un nouveau utilisateur dans la machine serveur par la commande **adduser** nom utilisateur pour créer un nouveau utilisateur

etape10 : teste si le service telnet marche dans l'intranet (réseau locale)

A partir d'une autre machine cliente qui appartient au réseau locale en essai de faire une connexion a distance à la machine serveur :

Telque @ip c'est l'adresse ip de la machine serveur (ou exécute le service telnet)

On exécute la commande suivante sur la machine cliente :

telnet @ip comme telnet @ip 23 # 23 numéro de port et @ip c'est l'adresse de la machine serveur ou tourne le service telnet

Etape 11 : tester le service telnet à partir de la machine cliente win7 virt en utilisant l'application putty.exe. Voila le programme :

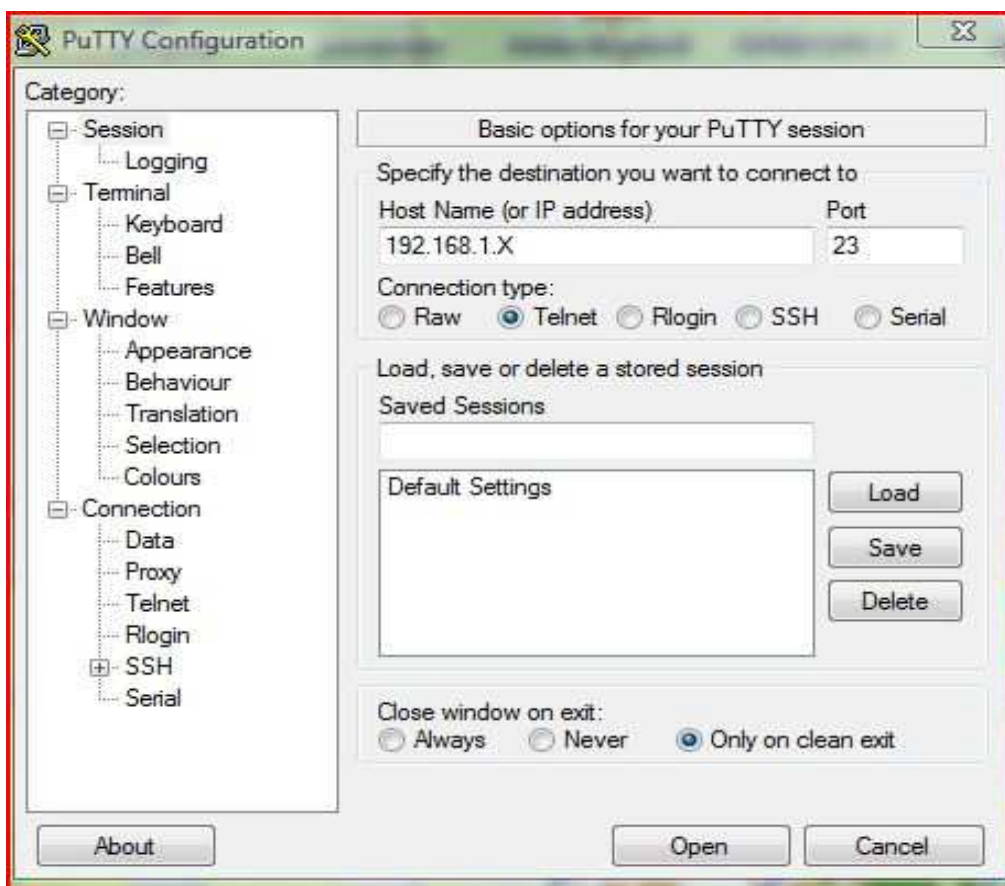


Figure 3.1 : putty

192.168.1.X : c'est l'adresse ip de la machine host serveur ou tourne le service telnet

Port : 23 c'est le port utilisé par le service telnet

3. Configuration serveur web Apache

Apache est un serveur http libre, c'est un des serveurs http les plus utilisés sur Internet avec plus de 60% des sites d'Internet.

un serveur http est un serveur hébergeant un ou plusieurs sites Web c'est à dire des pages html ou des programmes générant des pages html (programmes cgi) qui sont accessibles par des navigateurs internet. Le protocole, permettant l'échange de pages html est le protocole http, d'où le nom de serveur http. Ce protocole utilise généralement le port 80.

Pour lancer un serveur apache en écrit la commande : `# /etc/init.d/apache2 start.`

Pour vérifier que le démon apache tourne : `# ps -ef | grep apache`

3.1. Configuration de base

La configuration globale d'apache s'effectue par modification du fichier de configuration `/etc/apache2/apache2.conf`.

Les paramètres qui sont (en général) valables pour tous les serveurs se trouvent dans `apache2.conf` : **AccessFileName .htaccess**

Cette clause fixe le nom du fichier (par défaut `.htaccess`) à trouver dans un répertoire pour que l'accès de ce répertoire soit protégé, en imposant à l'utilisateur une authentification par nom et mot de passe.

Port 80

Apache écoute sur le port tcp 80

ServerRoot /etc/apache2

Il s'agit du répertoire où le serveur trouvera son répertoire de configuration.

3.2. Paramètres spécifiques à chaque serveur

Les paramètres (en général) spécifiques à chaque serveur (qui se trouvent dans `sites-enabled`)

DocumentRoot /var/www/html

Fixe la racine du serveur Web, c'est-à-dire le répertoire de base où sont recherchées par défaut les pages html, lorsque l'URL ne comporte pas de chemin de répertoire.

DirectoryIndex index.html index.php index.htm...

Il est courant d'omettre le nom du fichier de la page d'accueil d'un site ou de l'un de ses sous-répertoires. Pour ne pas retourner systématiquement une erreur 404 signalant une adresse erronée.

ServerAdmin webmaster@localhost

S'il a un problème, le serveur écrit un message à cette adresse

3.3. Contrôle des accès à un répertoire

Chaque répertoire auquel Apache accède peut être configuré, et root peut permettre certaines fonctionnalités d'apache pour ces répertoires, et en interdire d'autres.

Cela permet, en fonction des besoins et de la confiance accordée à chaque webmaster, de gérer les problèmes de sécurité.

En général, root cherche à donner tout juste les permissions qui sont requises en fonction des besoins. Le paramétrage d'un répertoire se précise dans un conteneur noté :

```
<Directory /chemin/vers/le/repertoire/> </Directory>
```

Exemple :

```
NameVirtualHost *
<VirtualHost *>
    DocumentRoot /home/monRepertoire/ # racine du site
    <Directory /> # droits du répertoire racine
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/monRepertoire/> # droits sur l'ensemble du site
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None # interdit les .htaccess
        Order allow,deny # donne l'ordre des permissions
        allow from all # autorise tous les clients
        # avec la directive suivante, il faut mettre la
        # page d'accueil dans /home/monRepertoire/apache2-default/
        # cela permet aux clients de se connecter avec
        # juste un http://monsite.com/
        RedirectMatch ~/$ /apache2-default/
    </Directory>

    etc...
</VirtualHost>
```

Principales options

Les principales options d'un répertoire peuvent être les suivantes :

None : Désactive toutes les options.

All : Active toutes les options SAUF MultiViews.

Indexes : Permet aux utilisateurs d'avoir des indexes générés par le serveur. C'est-à-dire si l'index du répertoire (index.htm le + souvent) est manquant, cela autorise le serveur à lister le contenu du répertoire (dangereux suivant les fichiers contenus dans ce répertoire).

FollowSymLinks : Autorise à suivre les liens symboliques.

ExecCGI : Autorise à exécuter des scripts CGI dans ce répertoire.

Donner les droits

Avec **Order allow,deny**, on peut permettre un accès à tous sauf quelques-uns. Par exemple,

```
Order allow,deny
allow from all # autorise tous les clients
deny from 192.168.0.67 # interdit l'accès par une IP
```

Permet à tous d'accéder sauf l'hôte 192.168.0.67.

Avec **Order deny, allow**, on peut permettre l'accès seulement par un sous-réseau. Par exemple,

```
Order deny,allow
Deny from all
Allow from 192.168.0
Allow from .mydomain.com
```

Permet l'accès seulement à partir du réseau local 192.168.0 et du domaine mydomain.com.

Directive AllowOverride

La directive AllowOverride permet au webmaster de redéfinir par lui-même certains droits ou certaines options spécifiquement dans certains répertoires. Pour cela, le webmaster crée dans un répertoire un fichier .htaccess dans lequel il définit les options et les droits qu'il souhaite. Par exemple, si root a mis dans les permissions d'un répertoire

3.4. Les commandes d'utilisation d'apache

Installation d'un serveur web apache2 : `sudo apt-get install apache2`

Démarrage du service `sudo /etc/init.d/apache2 start`

stoper 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 le 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 accès authentifié nous utilisons les deux fichiers suivants : .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 navigateurs Internet

Créer un site web appelé index.html

Créer un répertoire web dans le chemin /home/projet/

Mettre le site web index.html dans le répertoire web (/home/projet/web)

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

Exemple

```
<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/projet/web/"
    <Directory "/home/projet/web/">
        Options Indexes FollowSymLinks MultiViews

        AllowOverride AuthConfig
        Order allow,deny
        allow from all

    </Directory>
</VirtualHost >
```

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

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

<http://@ip-serveur-web/essai/index.html> <http://@ip-serveur-web/home/projet/web/index.html>

4. Configuration de un serveur de transfert des fichiers VsFTPd

De nombreux transferts de fichiers ont lieu à chaque instant sur internet. Le vieux protocole ftp (File Transfert Protocol) est toujours aussi utilisé parce qu'il est simple et rapide à mettre en place. Pour les utilisateurs, un transfert FTP est aujourd'hui facilité grâce à divers clients FTP totalement graphique comme FileZilla.

4.1. Installation et configuration

L'installation sous Ubuntu est comme toujours des plus simples : `$ sudo apt-get install vsftpd`

VsFTPd se configure via le fichier vsftpd.conf, positionné dans /etc sur la majorité des distributions.

Le fichier de configuration par défaut est très restrictif, il n'autorise que les connexions anonymes, en lecture seule. Il fait écouter le serveur sur toutes les interfaces disponibles, sur le port 21, et peut être tout à fait suffisant pour mettre en place un simple partage de fichier accessible à tous.

Démarrer le serveur FTP `sudo /etc/init.d/vsftpd restart`

arreter 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 ce connecté à leurs dossiers personnels, sans autoriser l'accès anonyme :

On indique la bannière

`ftpd_banner = Bienvenue sur le serveur Benmansour`

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 tous 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 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 utilise firefox (navigateur internet) ou client ftp :

ftp://@ip serveur ftp

Exemple : **ftp://192.168.1.65/**

192.168.1.65 adresses ip de serveur ftp

Accéder à un serveur FTP sous Windows

4.2. Utilisation le client ftp FileZilla

1- installez le client ftp FileZilla sur votre machine sous Windows et essayer de se connecter à distance a un serveur ftp qui se trouve sur une machine serveur ftp

Adresse ip de serveur ftp : 192.168.1.65

Port de serveur ftp : 21

Utilisateur : projet

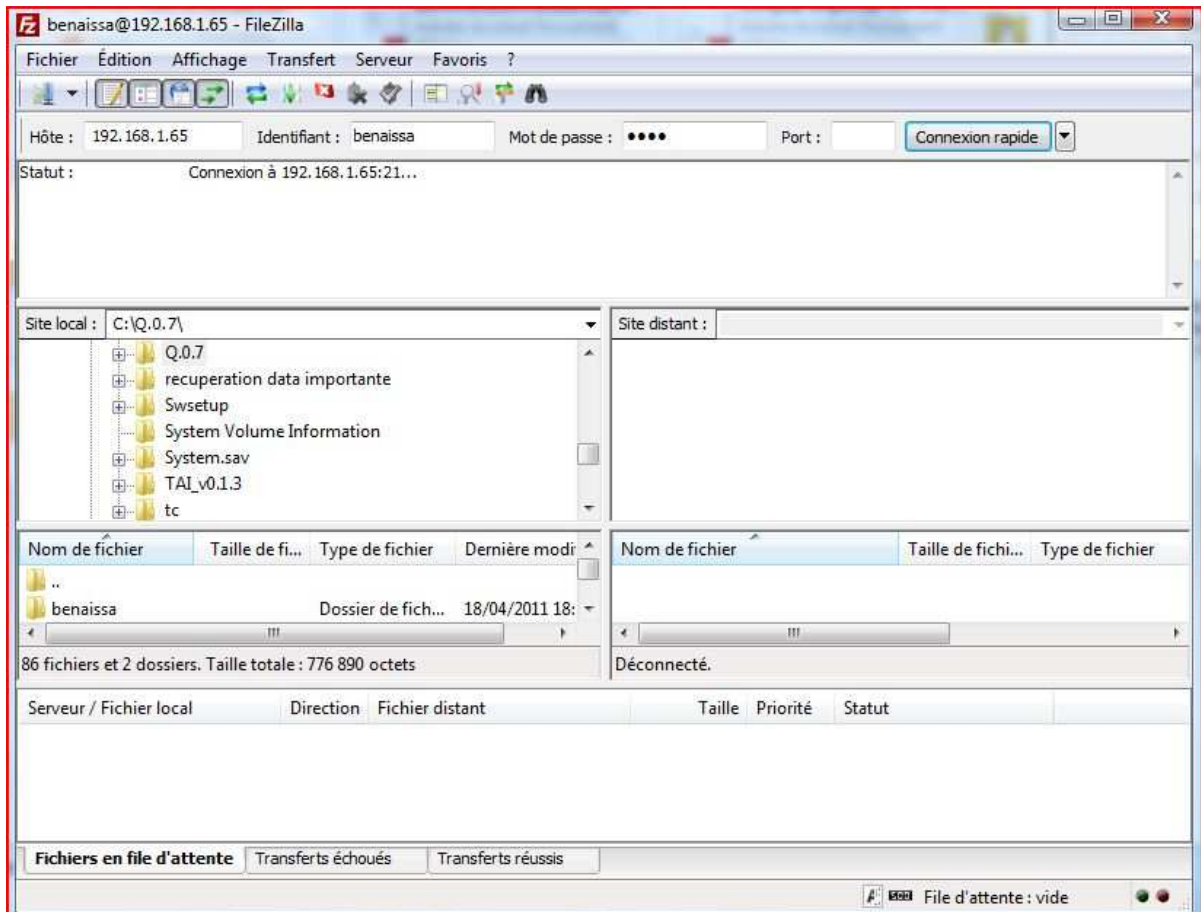


Figure 3.2 : filezilla

5. Le protocole SSH

Le protocole ssh (*Secure SHell*) est un protocole permettant à un client d'ouvrir une session Interactive sur une machine distante (serveur) afin d'administrer cette machine ou de transférer des fichiers de manière sécurisée

SSH est un protocole, c'est-à-dire une méthode standard permettant à des machines d'établir une communication sécurisée. A ce titre, il existe de nombreuses implémentations de clients et de serveurs SSH.

5.1 Etapes installation et configuration d'un protocole ssh

Etape 1 : Installation du client SSH

Sur le poste client (qui va prendre l'accès à distance) **openssh-client** installé par défaut sous Ubuntu doit être présent. Si le client `ssh` n'existe pas donc installé le paquet suivante : **openssh-client.deb**

Etape 2 : Installation du serveur SSH : à partir du paquet **openssh-server** sur pc serveur.

Démarrage le service : `sudo /etc/init.d/ssh start` ou `sudo service ssh start`

Pour l'arrêter le service : `sudo /etc/init.d/ssh stop` ou `sudo service ssh stop`

Pour redémarrage le service : `sudo /etc/init.d/ssh restart` ou `sudo service ssh restart`

Etape 3 : Copier des fichiers via SSH

Ou en termes profanes, si je désirais copier un fichier d'un de mes ordinateurs à l'autre, je procède de cette manière : `scp fichier.txt samir@192.168.1.65:/home/samir`

Ou copier : `scp -r repertoire samir@192.168.1.65:/home/samir/`

Etape 4 : Se connecter à un ordinateur distant via SSH

Pour ouvrir une session sur un ordinateur distant ayant un serveur SSH, vous devez écrire quelque chose comme ceci : `ssh <username>@<ipaddress>`
`ssh -l username ipaddress`

Exemple : `ssh samir@192.168.1.65`

Vous pouvez aussi appeler un ordinateur par son nom avec le domaine complet `ssh utilisateur@nom_machine.domain`

Exemple : `ssh samir@benaissa-ubuntu-fr.lan`

Etape 5 : Authentification par mot de passe

L'authentification par mot de passe (transmis chiffré) est le mode d'identification par défaut. Suite à l'installation du paquet **openssh-server** il peut parfois être nécessaire de modifier le fichier de configuration « `sshd_config` » notamment si vous rencontrez le problème suivant :
`moi@maison:~$ ssh user@domain.com Permission denied (publickey).`

Dans ce cas, il faut très basiquement modifier le fichier « `/etc/ssh/sshd_config` » de la manière suivante :

```
# Change to yes to enable tunnelled clear text passwords
PasswordAuthentication yes
```

Puis en cas de modifications, redémarrer le service avec la commande :
`sudo /etc/init.d/ssh restart`

Etape 6 : Le fichier de configuration du serveur SSH

AllowUsers samir benaissa

Ligne à ajouter, spécifie les *logins* des seuls utilisateurs (ici seuls samir et benaissa, pas mohamed) autorisés à se connecter.

PasswordAuthentication no

Passez de "yes" à "no" pour interdire l'utilisation du mot de passe et forcer l'usage de jeux de clefs public/privé (plus sûr).

Port 22 Spécifier le numéro de port de service SSH

ListenAddress 192.168.1.65 L'adresse IP d'écoute de serveur à distance (serveur de connexion).

Exemple de connexion avec authentification par clé publique et privé

Il faut que : **PasswordAuthentication no**

ssh root@192.168.1.65

Le ssh demande passphrase et non password

Le chemin de fichier de configuration est : `/etc/ssh/sshd_config`

Etape 7 : Connexion ssh sous windows

Utilisé le logiciel Putty sous Windows pour faire ouvrir une session sécurisé par ssh par :

- Entré le numéro de port de service SSH: 22
- Entré l'adresse ip de la machine distance (serveur SSH) : 192.168.1.65
- Essayé de faire une connexion par un post client à distance

6-Conclusion

Dans ce chapitre, nous avons installés les trois services réseau dans une machine virtuelle , c'est le serveur Telnet , le serveur web et le serveur de transfert les fichiers vsftp.

Les différents services installés et configurés dans ce chapitre seront utilisés dans la simulation et le test de différentes règles de notre firewall.

Dans le chapitre suivant, nous testons les différentes règles de firewall sur notre réseau virtuel.

1. Introduction

Dans ce chapitre, nous présentons les différents règles de notre pare feu afin de protégé notre réseau virtuel. La simulation de scénario de filtrage et sécurité est testé sur les différents services réseau comme le serveur apache, le serveur ftp et le serveur de connexion à distance telnet. Dans notre cas, nous avons créés 3 machine virtuelles a l aide de virtualbox.

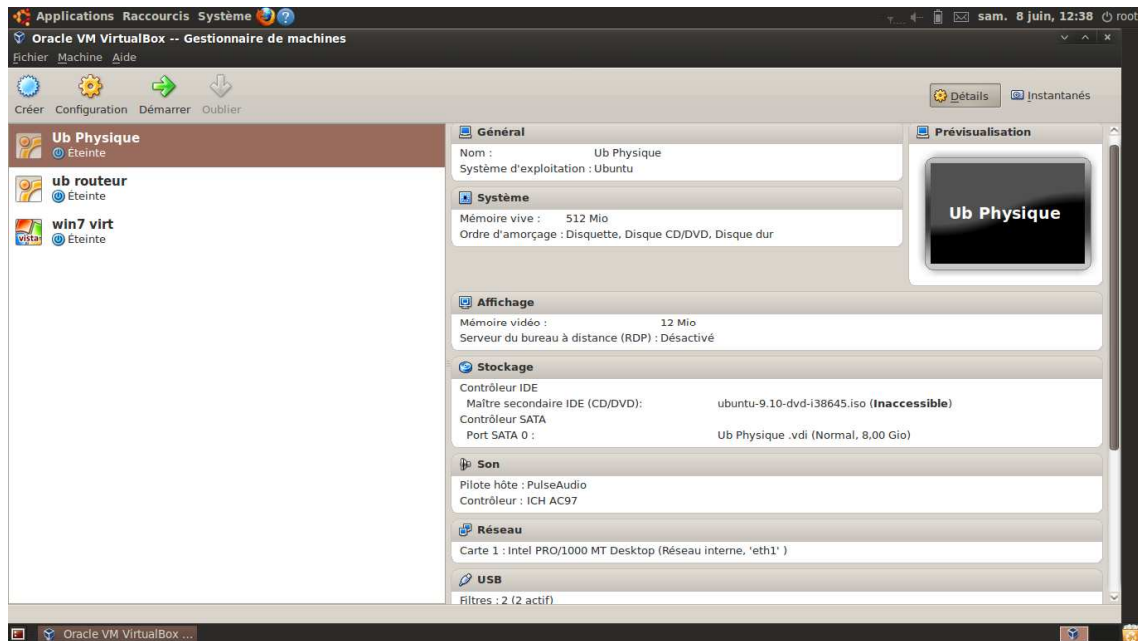
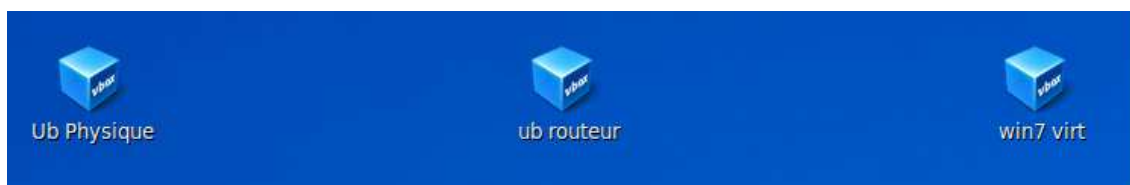


Figure 4.1 : Virtualbox

2. La conception de notre réseau virtuel

La 1^{er} machine est considéré comme une machine physique ubuntu 9.10 dans le rôle d un serveur. 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 2em machine virtuel ubuntu 9.10 dans le rôle d un routeur. Dans la configuration de cette machine on active 2 carte réseau interne la 1er on la nomme « eth0 » relié a la 3em machine, et la 2em « eth1 » qui relie le routeur avec le serveur. 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 »).



Après la création des machines on doit faire la liaison entre eux par les adresse IP
1er machine : ub physique IP: 192.168.2.2/24 avec une passerelle: 192.168.2.1
La 3em machine: win7 virt IP: 192.168.1.4/24 avec une passerelle: 192.168.1.1
La 2em machine: ub routeur a 2 adresse IP ce sont les passerelle utiliser 192.168.1.1 et 192.168.2.1.



La 1^{er} étape est accomplie on a fait les configurations nécessaires ; la 2em consiste à installer tout les paquets nécessaires au machines. Sur la machine physique on installe les différents serveurs : TELNET, SSH, FTP, WEB_APACHE.

On installe sur la machine client Win7 virtuelle : FILEZILLA (client ftp) et Putty (client telnet). PUTTY (émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet).

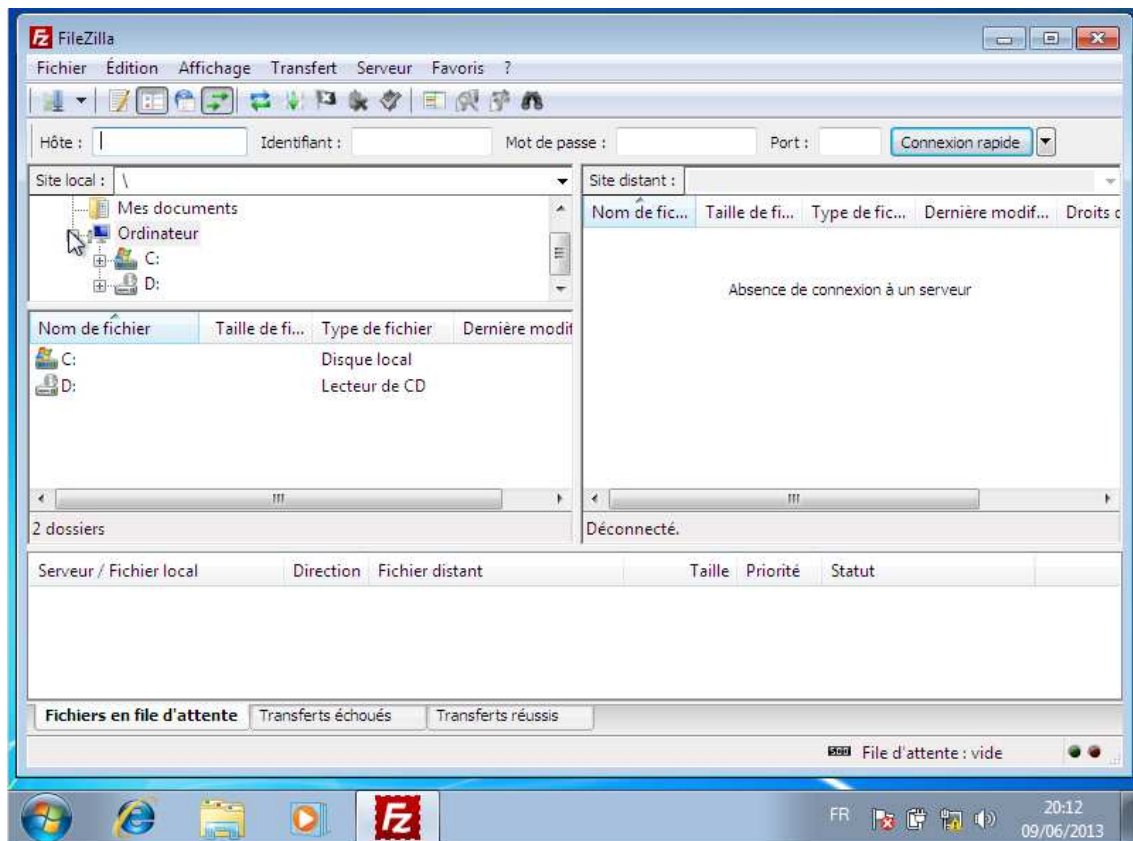


Figure 4.2 : filezilla

Pour y arriver on a qu'ajouter dans la configuration de nos 2 machines la détection d'une clé USB qui contient nos paquets.

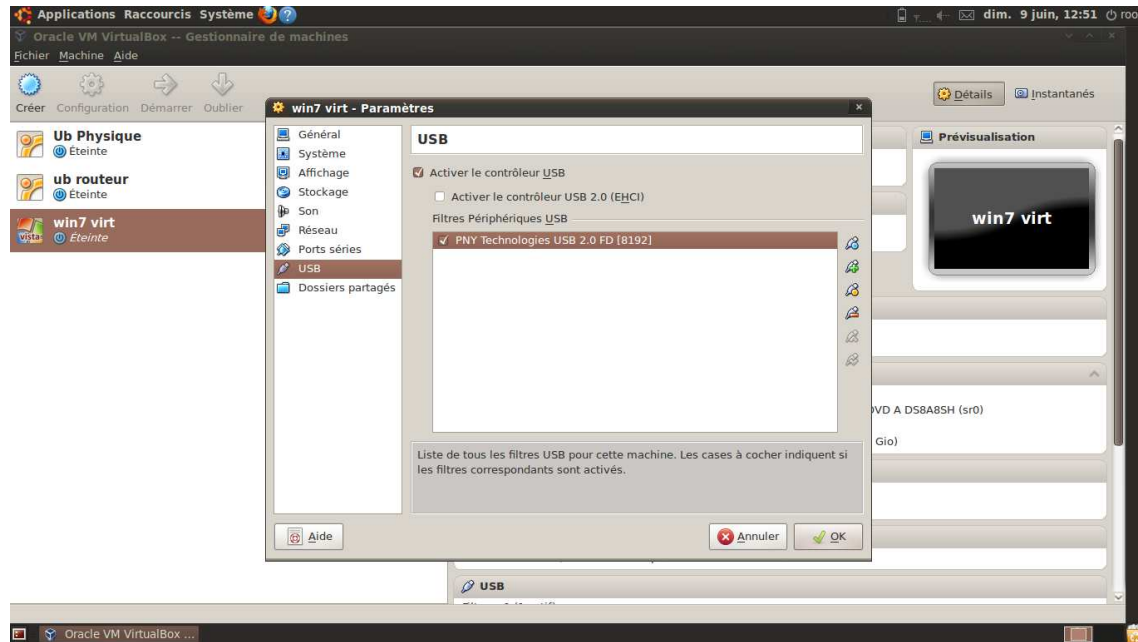


Figure 4.3 : l'ajout d'une clé USB virtuelle

A ce stade, la protection est nulle. Il n'y a pas de sécurité entre ses machines. C'est à dire que les machines peuvent faire tous les changements sans interaction. Piratage, Hackage... Tout est permis car la politique de chaque machine est en ACCEPT.

3. Architecture complète de notre réseau virtuel avec la configuration des cartes réseaux

Notre réseau virtuel est composé par 3 machines virtuelles comme suite :

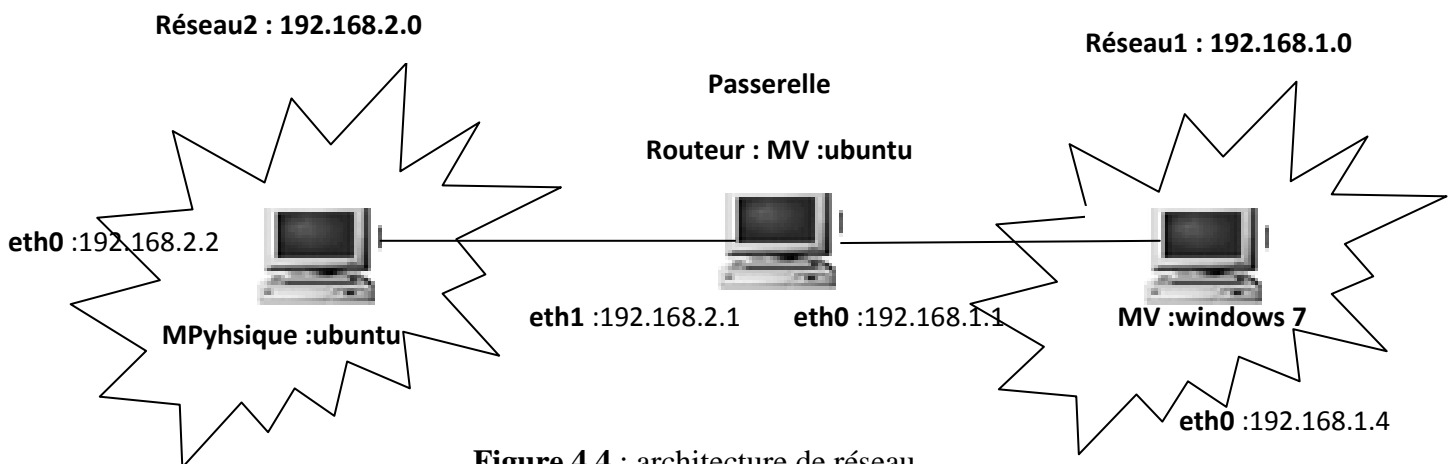


Figure 4.4 : architecture de réseau

Machine virtuelle ubuntu 9.10 : routeur ou passerelle entre le réseau 192.168.1.0 et 192.168.2.0

Dans la machine virtuelle nous avons créés deux cartes réseaux virtuel :
eth0 : 192.168.1.1 et eth1 : 192.168.2.1

Dans le fichier /etc/network/interfaces nous avons ajouté les lignes suivantes :

```
auto eth0
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
```

```
auto eth1
iface eth1 inet static
address 192.168.2.1
netmask 255.255.255.0
```

Activation la propriété de routage de la passerelle:

Dans le fichier /etc/sysctl.conf nous avons activés la ligne suivante (enlever la #) :
net.ipv4.ip_forward=1

Machine virtuelle hôte physique ubuntu :

Ajouter la configuration suivante dans le fichier /etc/network/interfaces :

```
auto eth1
iface eth1 inet static
address 192.168.2.2
netmask 255.255.255.0
up route add -net 192.168.1.0/24 gw 192.168.2.1
```

machine virtuelle xp :

Sur la machine virtuelle : paramètre : configuration réseau

Nous avons ajoutés une carte réseau1 eth0:

Nous avons choisis réseau interne (nommé votre carte réseau par eth0)

Nous avons démarrés la machine virtuelle xp et nous avons configurés l'adresse ip de notre machine

exemple @ip : 192.168.1.4

mask : 255.255.255.0

passerelle : 192.168.1.1 (passerelle de machine virtuelle ubuntu)

Nous avons aussi désactiver le pare-feu de Windows 7

Teste de la configuration entre les trois machines avec le ping :

Machine physique ubuntu : @ip: 192.168.2.2

ping 192.168.1.1 (carte réseau1 de passerelle machine virtuelle ubuntu)

ping 192.168.2.1 (carte réseau de passerelle machine virtuelle ubuntu)

ping 192.168.1.4 (carte réseau de machine virtuelle xp)

Machine virtuelle ubuntu : passerelle :
ping 192.168.1.4 (machine virtuelle xp)
ping 192.168.2.2 (machine physique hôte ubuntu)
ping 192.168.1.1 (carte réseau1 de passerelle)
ping 192.168.2.1 (carte reseau2 de la passerelle)

machine virtuelle xp : @ip : 192.168.1.4
ping 192.168.2.2 (machine physique hôte ubuntu)
ping 192.168.1.1 (carte réseau1 de passerelle virtuelle ubuntu)
ping 192.168.2.1 (carte réseau2 de passerelle virtuelle ubuntu)

Si on a 3 machines virtuelles alors la configuration est comme suite :
Machine virtuelle 1 : réseau interne (eth0)
Machine virtuelle 2 : réseau interne (eth0)
Machine virtuelle 3 : réseau interne (eth0)

4. Les types de connexions réseaux avec Oracle VirtualBox

L'utilisation des machines virtuelles avec Oracle VirtualBox nécessite une configuration adéquate de notre réseau virtuel. Nous distinguons plusieurs « type » de connexions et de configuration réseau.

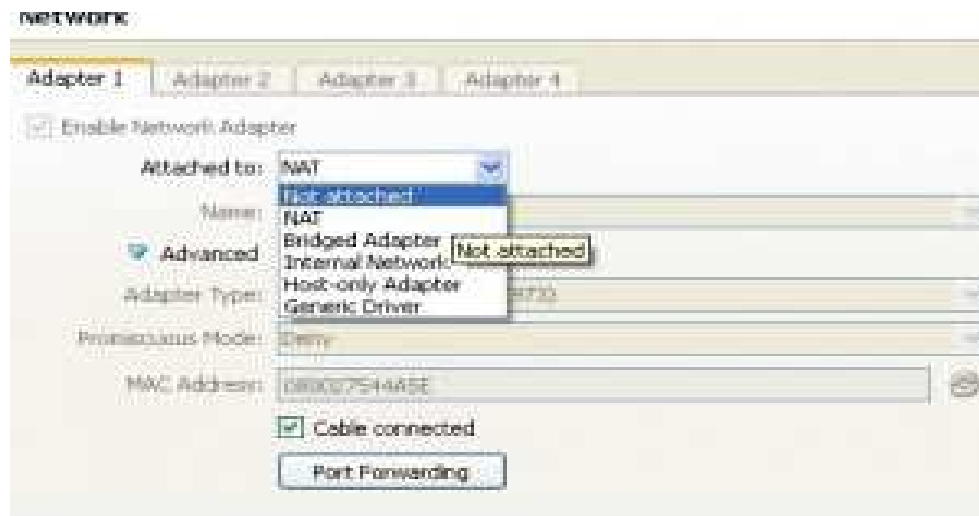


Figure 4.5 : connexion réseau virtualbox

- ✓ **Pour le type « NAT »**
 - Les Machines Virtuelles communiquent entre elles
 - Les Machines Virtuelles communiquent avec l'hôte et l'extérieur
 - L'hôte et l'extérieur ne voient pas les VM

Dans ce type, les trames allant vers l'extérieur de votre machine virtuelle auront la même adresse que votre machine hôte (peut importe l'adresse IP de votre machine virtuelle).

Particularité : Dans ce mode, la machine virtuelle ne peut être utilisée qu'en client. Elle ne peut pas recevoir de requêtes directes de l'extérieur (ex : un ping ne fonctionnera pas).

✓ **Pour le type « Réseau Interne » :**

- Les Machines Virtuelles communiquent entre elles
- Les Machines Virtuelles ne communiquent pas avec l'hôte
- Les Machines Virtuelles ne communiquent pas avec l'extérieur

Ce type permet de connecter des machine virtuelle entre-elles sur un réseau virtuel isolé.

✓ **Pour le type : « Réseau Privé Hôte »**

- Les Machines Virtuelles communiquent entre elles
- Les Machines Virtuelles communiquent avec l'hôte
- Les Machines Virtuelles ne communiquent pas avec l'extérieur

Avec ce type de connexion réseau votre machine virtuelle ne peut communiquer qu'avec votre machine hôte de la même manière qu'avec deux cartes physiques standard.

• **Pour le type « Pont » :**

- Les Machines Virtuelles sont sur le même réseau que l'hôte

Avec ce type de connexion, les trames qui sortent de votre machine virtuelle auront leurs particularités propres (adresse MAC et adresse IP).

5. Les règles de configuration d'un pare-feu

IPtables (associé à Netfilter) est un des meilleurs firewalls pour Linux, et certainement le plus répandu. L'objectif de notre travail est de configurer ce firewall
Utilisez la commande **iptables -L -v** pour lister les règles en place.

Arguments utilisés :

- i : interface d'entrée (input)
- o: interface de sortie (output)
- t : table (par défaut *filter* contenant les chaînes INPUT, FORWARD, OUTPUT)
- j : règle à appliquer (Jump)
- A : ajoute la règle à la fin de la chaîne (Append)
- I : insère la règle au début de la chaîne (Insert)
- R : remplace une règle dans la chaîne (Replace)

- D : efface une règle (Delete)
- F : efface toutes les règles (Flush)
- X : efface la chaîne
- P : règle par défaut (Policy)
- lo : localhost (ou 127.0.0.1, machine locale)

Commandes iptables

La commande iptables permet d'ajouter, de modifier ou de supprimer des règles de filtrage :

	TABLE	CHAINE	MOTIF(S)	CIBLE
iptables	-t filter	-A INPUT	-p <protocole>	-j ACCEPT
	nat	OUTPUT	-s <@ sce>	DROP
	mangle	FORWARD	-d <@ dest>	DENY
		PREROUTING	--sport <port sce>	MASQUERADE
		POSTROUTING	--dport <port dest>	SNAT
			-i <if entrée>	DNAT
			-o <if sortie>	REDIRECT

6. Exemple de scénario de règles pour étudier le filtrage et la sécurité de notre réseau virtuel

Pour régler le problème de sécurité on doit entrer des règles à respecter dans notre système (à l'aide Iptables).

On a fait une simulation d'un scénario de filtrage et sécurité qui nécessite d'abord qu'on a déjà fait les étapes précédentes y compris installation de tous les paquets nécessaires au bon déroulement de notre mise en forme de pare-feux.

6.1. Filtrage et sécurisation sur la table filter

Par défaut la tables traites sans une précision d une tables voulus est le Filter. Pour afficher les chaine existantes de cette table on utilise = iptables -L .pour plus de détails iptables -L -v.

Si on veut savoir que la politique des chaines iptables -S

Pour la changer dans la chaine input en peut y arriver par 2 méthodes

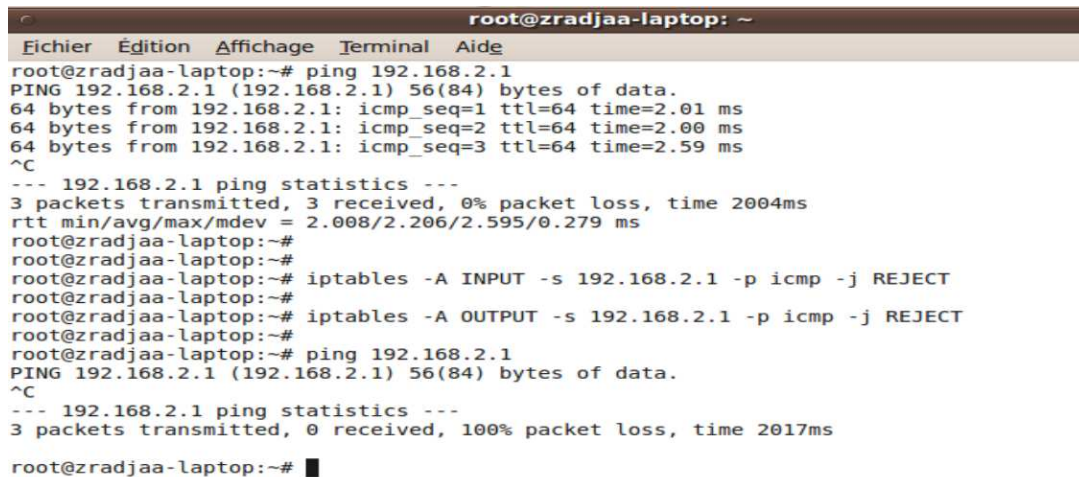
Iptables -P INPUT DROP ou bien iptables -A INPUT -j DROP

Sur la machine serveur 192.168.2.2 :

On veut écrire une règle qui permet de refuser le ping entre cette machine et le routeur

Iptables -A INPUT -s 192.168.2.1 -p icmp -j REJECT

Iptables -A OUTPUT -s 192.168.2.1 -p icmp -j REJECT



```
root@zradjaa-laptop: ~
Fichier Édition Affichage Terminal Aide
root@zradjaa-laptop:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=2.01 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=2.00 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=2.59 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.008/2.206/2.595/0.279 ms
root@zradjaa-laptop:~#
root@zradjaa-laptop:~#
root@zradjaa-laptop:~# iptables -A INPUT -s 192.168.2.1 -p icmp -j REJECT
root@zradjaa-laptop:~#
root@zradjaa-laptop:~# iptables -A OUTPUT -s 192.168.2.1 -p icmp -j REJECT
root@zradjaa-laptop:~#
root@zradjaa-laptop:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2017ms
root@zradjaa-laptop:~# █
```

Figure 4.6 : Le teste du ping sur « ub physique ».

Après réflexion on veut l'autoriser mais comme ordre est important on ne peut pas ajouter une nouvelle règle qui autorise le ping car celle si vas ajouter en dernier. Donc 3 suggestions sont possibles :

Soit faire une insertion et garder l'ancienne règle :

iptables -I INPUT 1 -p icmp -j ACCEPT

Soit remplacer l'ancienne règles par la nouvel *iptables -R INPUT 1 -p icmp -j ACCEPT*

Ou bien supprimer l'ancienne règle est ajouté une nouvelle règle

Iptables -D INPUT 1 Ou bien *iptables -D INPUT -p icmp -j REJECT*

Iptables -A INPUT -p icmp -j ACCEPT

Puis répéter les règles pour la chaine output a la place input.

on veut accepter tout les paquets venant du win7 virtuelle :

iptables -A input -s 192.168.1.4/24 -p tcp -j ACCEPT

Interdire la machine client win7 virt d accéder au web apache :

```
iptables -A INPUT -s 192.168.1.4/24 -p tcp --dport 80 -j DROP
```

```
iptables -A OUTPUT -s 192.168.1.4/24 -p tcp --dport 80 -j DROP
```

Interdire la connexion ftp de la machine client ubuntu (ub routeur):

```
iptables -A INPUT -s 192.168.2.1/24 -p tcp --dport 20:21-j REJECT
```

-Sur la machine routeur :

Interdire le routage des paquets entre win7 virt et ub physique :

```
iptables -P FORWARD DROP
```

Autoriser la machine client win7 virtuelle d'accéder au serveur telnet qui est sur la machine serveur ub physique :

```
iptables -A FORWARD -s 192.168.1.4/24 -d 192.168.2.2/24 -p tcp --dport 23 -j ACCEPT
```

Interdire la machine Windows 7 d'interroger le serveur DNS par le forwarding :

```
iptables -t filter -A FORWARD -s 192.168.1.4 -d 192.168.2.2 -p tcp --dport 53 -j DROP
```

```
iptables -t filter -A FORWARD -s 192.168.1.4 -d 192.168.2.2 -p udp --dport 53 -j DROP
```

Autoriser win7 virtuelle d accéder au serveur ftp par le routage :

```
iptables -t filter -A FORWARD -s 192.168.1.4 -d 192.168.2.2 -p tcp --dport 20:21 -j ACCEPT
```

Autoriser loopback : **iptables -t filter -A INPUT -i lo -j ACCEPT**

```
iptables -t filter -A OUTPUT -o lo -j ACCEPT
```

Pour tester on exécute le ping sur la machine voulue.

Accepter l'entrée et sortie de la demande d SSH:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
```

Autoriser la sortie d NTP :

```
iptables -A OUTPUT -p udp --dport 123 -j ACCEPT
```

Ne pas casser les connexions établies :

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -i eth1 -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
```

6.2. Filtrage et sécurisation sur la table nat

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.2.2 --dport 80 -j DNAT  
--to-destination 192.168.1.1-192.168.1.4
```

```
iptables -t nat -A POSTROUTING -p tcp -d 192.168.2.2 --dport 80 -j SNAT \  
--to-source 192.168.1.4
```

```
iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000
```

```
iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source 192.168.1.4-  
192.168.2.2 :1024-32000
```

6.3. Autres règles testées sur le réseau virtuel (exemple réalisé) :

Vider les tables actuelles

```
iptables -t filter -F
```

Vider les règles personnelles

```
iptables -t filter -X
```

Interdire toute connexion entrante et sortante

```
iptables -t filter -P INPUT DROP
```

```
iptables -t filter -P FORWARD DROP
```

```
iptables -t filter -P OUTPUT DROP
```

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
```

Autorisé loopback (connection localhost)

```
iptables -t filter -A INPUT -i lo -j ACCEPT
```

```
iptables -t filter -A OUTPUT -o lo -j ACCEPT
```

Service ping: ICMP (Ping)

```
iptables -t filter -A INPUT -p icmp -j ACCEPT
```

```
iptables -t filter -A OUTPUT -p icmp -j ACCEPT
```

Service secure shell input : SSH

```
iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
```

Service secure shell output : SSH

```
iptables -t filter -A OUTPUT -p tcp --dport 22 -j ACCEPT
```

serveur de resolution de nom DNS

```
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
```

Serveur de temps (ntp : network time protocole)

```
iptables -t filter -A OUTPUT -p udp --dport 123 -j ACCEPT
```

Serveur web HTTP + HTTPS Output

```
iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT
```

Serveur web HTTP + HTTPS Input

```
iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 8443 -j ACCEPT
```

Serveur de transfert des fichiers FTP Output

```
iptables -t filter -A OUTPUT -p tcp --dport 20:21 -j ACCEPT
```

Serveur de transfert des fichiers FTP Input

```
iptables -t filter -A INPUT -p tcp --dport 20:21 -j ACCEPT
iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Serveur de messagerie électronique 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
```

Serveur de récupération 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
```

Serveur de récupération 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
```

Serveur de récupération 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
```

7. Conclusion

Le firewall propose un véritable contrôle sur le trafic réseau. Il permet d'analyser, de sécuriser et de gérer le trafic réseau, et ainsi d'utiliser le réseau de la façon pour laquelle il a été prévu. Tout ceci sans l'encombrer avec des activités inutiles, et d'empêcher une personne sans autorisation d'accéder à ce réseau de données.

Le firewall c'est moyen principale dans la sécurité et la protection de notre réseau. Cet outil de contrôle et protection à été testé sur un réseau composé par des machines virtuelles créés par virtualbox.

Conclusion générale

Un pare-feu est un système permettant de protéger un ordinateur ou un réseau d'ordinateurs des intrusions provenant d'un réseau tiers (notamment internet). Le pare-feu est un système permettant de filtrer les paquets de données échangés avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum des interfaces réseau.

Chaque ordinateur connecté à internet (et d'une manière plus générale à n'importe quel réseau informatique) est susceptible d'être victime d'une attaque d'un pirate informatique. La méthodologie généralement employée par le pirate informatique consiste à scruter le réseau (en envoyant des paquets de données de manière aléatoire) à la recherche d'une machine connectée, puis à chercher une faille de sécurité afin de l'exploiter et d'accéder aux données s'y trouvant.

Le firewall donc offre au système une protection très efficace contre les dangers de l'Internet. Son installation au sein d'un réseau permet de limiter l'étendue des dégâts potentiels et d'enrayer les attaques avant qu'elles ne compromettent l'intégrité de l'ensemble du réseau.

Comme on peut le constater, les firewalls possèdent de multiples capacités qui peuvent différer en fonction de leurs types. Cette multitude de solutions impose donc une étude rigoureuse de la sécurité devant être mise en place.

Les recherches pour faire évoluer les technologies de filtrage sont nées du besoin de sécuriser les échanges réseaux. Pour améliorer ce filtrage il a été nécessaire de remonter dans les couches OSI, ce qui a été rendu possible grâce à une technologie logicielle et matérielle de plus en plus rapide.

Néanmoins, le firewall représente un goulet d'étranglement face aux débits sans cesse croissants, et aux applications de plus en plus compliquées qui surchargent de travail le firewall. On peut donc se demander si les performances de débit d'un firewall pourront évoluer aussi vite que les performances des réseaux haut-débit ou assisterons-nous à une dégradation du service firewall au profit d'un meilleur débit ?

Dans notre projet de fin d'étude, nous avons essayés de configurer et tester les différentes règles de firewall sur un réseau virtuel créé par virtualbox.

Bibliographie

[1] <http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/index.html>.

[2] <http://www.cru.fr/securite/CRUGB/principe.html>.

[3] http://www.reseaux-telecoms.net/articles_btree/189_9/Article_view
Article du 05/10/2001 : Portes coupe-feu : marché ouvert, réseau fermé.

[4] <http://www.iperformances.fr/public/FireProof.html>.

[5] <http://www.scssi.gouv.fr/document/fiches/fire-fr.html>.

[6] <http://pic.dhe.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cmds%2Fdoc%2Faixcmds3%2Finetd.htm>.

[7] <http://www.udivers.com>.

Listes des figures

- **Chapitre N 1 : Introduction a la virtualisation**

- **Figure 1.1** : Virtualisation
- **Figure 1.2** : Hyperviseur de type 1
- **Figure 1.3** : Hyperviseur de type

- **Chapitre N 2 : Notion de base d'un pare-feu**

- Figure 2.1** : Le filtrage de paquet
- Figure 2.2** : Firewalls de niveau circuit
- Figure 2.3** : Filtrage dynamique de paquets
- Figure 2.4** : Firewall avec routeur de filtrage
- Figure 2.5** : Passerelle double ou réseau bastion
- Figure 2.6** : Firewall avec réseau de filtrage
- Figure 2.7** : Firewall avec sous-réseau de filtrage
- Figure 2.8** : Fonctionnement d'un firewall
- Figure 2.9** : Iptables
- Figure 2.10** : Firwall
- Figure 2.11** : Table NAT

- **Chapitre N 3 : Installation et configuration des différents services réseaux**

- Figure 3.1** : Putty
- Figure 3.2** : Filezilla

- **Chapitre N 4 : Simulation d'un scénario de filtrage et sécurité sur un réseau virtuel**

-Figure 4.1 : Virtuelbox

-Figure 4.2 : Filezilla

-Figure 4.3 : l'ajout une clé USB virtuel

-Figure 4.4 : Architecture de réseau

-Figure 4.5 : Connexion réseau virtualbox

-Figure 4.6 : Le teste du ping sur « ub physique ».

Résumé :

Un firewall offre au système une protection du réseau interne très efficace contre les dangers de l'Internet. Stop les dégâts potentiels et les attaques, tout en permettant de travailler sans trop de contraintes. Ceci est possible grâce à des techniques de filtrage rapides et intelligentes.

Sur le système Linux nous Avons configuré et tester les différentes règles de firewall sur un réseau virtuel créé par virtualbox avec l'installation et la configuration des différents services réseau (SSH, Telnet, FTP, WEB-Apache) sur la machine serveur et des services clients pour les testé (Putty, Filezilla) sur la machine virtuel client Windows.

Mots clés : firewall, serveur telnet, serveur apache, serveur vsftp, serveur SSH, iptables, netfiltre, table NAT, sécurité, protection, filtrage

Abstract :

A firewall system provides a very effective protection of internal network against the dangers of the Internet. Stop potential damage and attacks, while allowing to work without too many constraints. This is possible due to technical fast and smart filtering.

On Linux system we Have configured and test different firewall rules on a virtual network created by virtualbox with the installation and configuration of various network services (SSH, Telnet, FTP, WEB-Apache) on the server machine and services customers to test (Putty, Filezilla) on the Windows client virtual machine.

Key words : firewall, telnet server, apache server, vsftp server, SSH server, iptables, netfiltre, NAT table, security, protection, filtering

ملخص:

نظام جدار الحماية يوفر حماية فعالة جدا من الشبكة الداخلية ضد أخطار الإنترنت. يوقف الأضرار والهجمات المحتملة في حين يسمح العمل دون قيود كثيرة جدا، و هذا ممكن بتقنيات التصفية السريعة والذكية.

بنظام Linux قمنا بتكوين واختبار قواعد جدار الحماية مختلفة على الشبكة الافتراضية التي أنشأناها بـ Virtualbox مع التركيب والتكوين بخدمات الشبكة المختلفة (SSH, Telnet, FTP, Web-Apache) على جهاز الخادم والخدمات (Putty, FileZilla) لاختبار الجهاز الوهمي عميل الويندوز.

الكلمات المفتاحية: جدار الحماية، الخادم Telnet، الخادم Apache، الخادم SSH، والخادم vsftp، جداول الIP، صافي الترشيح، جدول NAT، الأمن والحماية والترشيح
