La sécurité informatique

Projet de :

LABHAIRI MEHDI &YOUNESBOUIRANE

Introduction

La sécurité informatique, un domaine prometteur encore en plein essor dans le monde informatique.
Notre projet porte sur les firewall. Il nous a été attribué dans le cadre de la politique d'amélioration des connaissances des étudiants en fin de cycle. En effet chaque année les étudiants en fin de cycle (deuxième année) doivent en étroite collaboration avec le corps professoral choisir un projet qu'il développeront.
Ainsi nous avons eu comme projet la sécurité et les firewall. Par ce projet nous avions pour but d'acquérir un maximum de connaissance théorique et pratique sur la sécurité informatique en général et les firewalls en particulier .l'école supérieure de technologie disposant d'un architecture réseaux avec un certains nombre de matériel, nous avons trouvé plus intéressant de nous baser sur cette architecture réseaux.
Pour le bon déroulement de notre projet nous avons décidé depuis le début du temps que nous utiliserons pour élaborer ce travail qui n'ai point chose facile si et qui demande que nous ménagions nos efforts.

A. Aspects généraux de la sécurité

1. Les attaques réseaux

Avec l'avènement d'Internet et des moyens de communication modernes, une nouvelle forme d'insécurité est apparue, qui s'appuie sur l'utilisation de code informatique pour perturber ou pénétrer les systèmes informatiques.
Les attaques touchent généralement les trois composantes suivantes d'un système : la couche réseau, en charge de connecter le système au réseau, le système d'exploitation. En charge d'offrir un noyau de fonction au système, et la couche application, en charge d'offrir des services spécifiques.
Toutes ces composantes d'un système constituent autant de vecteurs de pénétration pour des attaques de toute nature (les attaques visant les systèmes d'exploitation et les applications ne sont pas détaillées).
Aujourd'hui, ces attaques sont si nombreuses qu'il serait illusoire de prétendre es décrire toutes. Il est cependant important de dresser une typologie des attaques réseaux à base des faiblesses de sécurité afin de mieux les cerner en majorité.
Comme tout effet a une cause, les attaques réseau s'appuient sur divers types de faiblesses, que l'on peut classifier par catégorie :

- Faiblesses des protocoles.
- Faiblesses d'authentification.
- Faiblesses d'implémentation ou bogues.
- Mauvaises configurations.

En s'appuient sur ces faiblesses, le pirate peut lancer un ensemble d'attaques afin de récolter des informations importantes mais aussi de pénétrer un système réseau.

2. Conduire une politique de sécurité réseau

Après avoir présenté, à la partie précédente, les menaces qui pèsent sur le réseau d'entreprise, nous décrivons dans cette partie les étapes nécessaires pour définir une politique de sécurité réseau et élaborer des stratégies de sécurité autour de cette politique.

2.1. Définir une politique de sécurité réseau

Objectifs d'une politique de sécurité réseau

La définition d'une politique de sécurité n'est pas un exercice de style mais une démarche de toute entreprise visant à protéger son personnel et ses biens d'éventuels incidents de sécurité dommageables pour son activité.
La définition d'une politique de sécurité réseau fait intégralement partie de la démarche sécuritaire de l'entreprise, elle s'étend à de nombreux domaines, dont les suivants :

- Audit des éléments physiques, techniques et logiques constituant le système d'information de l'entreprise.
- Sensibilisation des responsables de l'entreprise et du personnel aux incidents de sécurité et aux risques associés.
- Formation du personnel utilisant les moyens informatiques du système d'information.
- Structuration et protection des locaux abritant les systèmes informatiques et les équipements de télécommunications, incluant le réseau et les matériels.
- Gestion du système d'information de l'entreprise lui permettant de suivre et d'appliquer les recommandations sécurité des procédures opérationnelles.
- Définition du cadre juridique et réglementaire de l'entreprise face à la politique de sécurité et actes de malveillance.
- Classification des informations de l'entreprise selon différents niveaux de confidentialité et de criticité.
- Les objectifs d'une politique de sécurité réseau, qui, rappelons-le, viennent en complément de la politique générale de sécurité de l'entreprise, sont identiques et utilisent les mêmes concepts, notamment les suivants :

Définir une politique de sécurité réseau

Une politique de sécurité réseau doit respecter un ensemble de principes génériques :

- **Identification** : Information permettant d'indiquer qui vous prétendez être. Une identification élémentaire est le nom d'utilisateur que l'on saisit dans un système informatique. Une identification plus évoluée peut être le relevé d'empreinte digitale, l'analyse fiscale, rétinienne, etc.

- **Authentification** : Information permettant de valider l'identité pour vérifier que vous êtes celui que vous prétendez être. Une authentification élémentaire est le mot de passe que vous entrez dans le système informatique. Une authentification forte combine une chose que vous possédez et une chose que vous connaissez, (numéro de cane bancaire et code personnel, par exemple).
- **Autorisation** : Information permettant de déterminer quelles seront les ressources de l'entreprise auxquelles l'utilisateur identifié et autorisé aura accès ainsi que les actions autorisés sur ces ressources. Cela couvre toutes les ressources de l'entreprise.
- **Confidentialité** : Ensemble des mécanismes permettant qu'une communication de données reste privée entre un émetteur et un destinataire. La cryptographie ou le chiffrement des données est la seule solution fiable pour assurer la confidentialité des données.
- **Intégrité** : Ensemble des mécanismes garantissant qu'une information n'a pas été modifiée.
- **Disponibilité :** Ensemble des mécanismes garantissant que les ressources de l'entreprise sont accessibles, que ces derniers concernent l'architecture réseau, la bande passante, le plan de sauvegarde, etc.
- **Nom répudiation :** Mécanisme permettant de garantir qu'un message a bien été envoyé par un émetteur et reçu par un destinataire.
- **Tracabilité :** Ensemble des mécanismes permettant de retrouver les opérations réalisées sur les ressources de l'entreprise. Cela suppose que tout événement applicatif soit archivé pour investigation ultérieure.

2.1.3. Les différents types de politique de sécurité

Une politique de sécurité peut être trop permissive ou au contraire trop restrictive.
Dans le premier cas, elle risque de présenter une faiblesse de sécurité par son coté laxiste. Dans le second cas, elle peut devenir inapplicable du de règles trop strictes.
Comme dans de nombreux domaines, seule l'expérience guide l'écriture d'une politique de sécurité ainsi que ces règles. Dans tous les cas, plus les ressources sont critiques, plus les règles doivent être strictes.
Quelle que soit la politique de sécurité défini, il faut savoir gérer les exceptions aux règles de sécurité. Ces exceptions sont connues, limitées, documentées et sous contrôle.

2.2 Stratégie de sécurité réseau

Après avoir défini les objectifs et le contenu d'une politique de sécurité réseau à la section précédente, nous détaillons dans cette partie les stratégie de sécurité à mettre en ouvre autour d'une telle politique afin de répondre de la manière la plus adaptée aux objectifs de sécurité.
L'établissement de stratégies de sécurité exige de prendre en compte l'historique de l'entreprise. L'étendue de son réseau, le nombre d'employés, la sous-traitance avec des tierces parties, le nombre de serveurs, l'organisation du réseau, etc. D'une manière générale une bonne stratégie de sécurité vise à définir et mettre en œuvre des mécanismes de sécurité, des procédures de réponse aux incidents de sécurité et des contrôles et audits de sécurité.

-Méthodologie pour élaborer une stratégie de sécurité réseau

Diverses méthodes permettent d'élaborer des stratégies de sécurité.

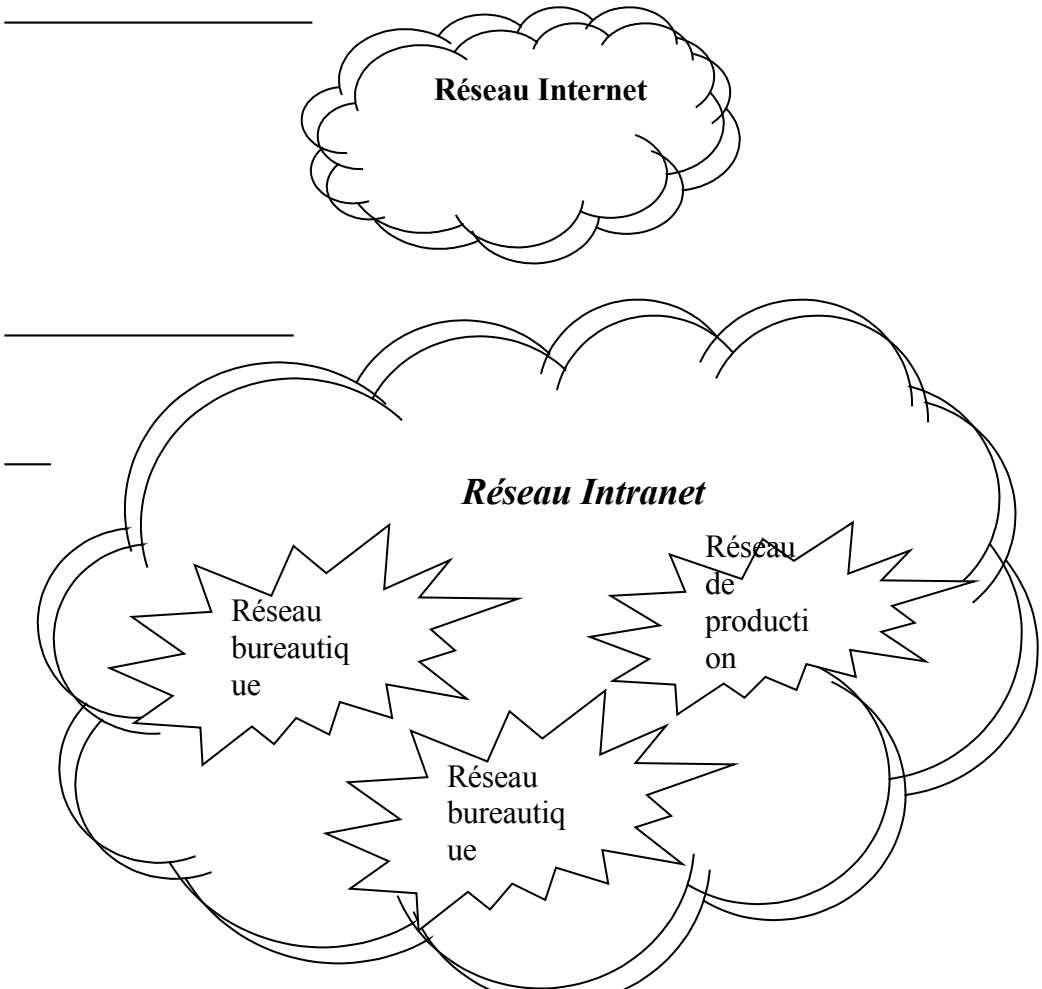
Stratégie de sécurité

- **Prédiction des agressions potentielles et analyse de risque**
  - Pour chaque type de menace (agresseur intentionnel, erreur, etc.)
    - Pour chaque méthode d'attaque (virus, etc.)
      - Stratégie proactive
        - Prédiction des dommages possibles
        - Détermination du degré de vulnérabilité
        - Minimisation de ce degré de vulnérabilité
          - Mise en œuvre et développement des Stratégies et contrôle de sécurité
        - Elaboration des plans de contingence
      - Stratégie réactive
        - Analyse des dommages

→ Détermination de leur origine
→ Réparation des dommages
↳ Mise en œuvre et développement des Stratégies et contrôle de sécurité
→ Mise en œuvre du plan de contingence

→ Analyse des résultats et exécution des simulations
→ Analyse de l'efficacité des stratégies
→ Modification des stratégies pour amélioration

-**Stratégie des goulets d'étranglement**



Les goulets d'étranglement

Principe

Des contrôles d'authentification sont mis en place afin d'authentifier les accès aux périmètres de sécurité.

Stratégies de séparation des pouvoirs

Principe

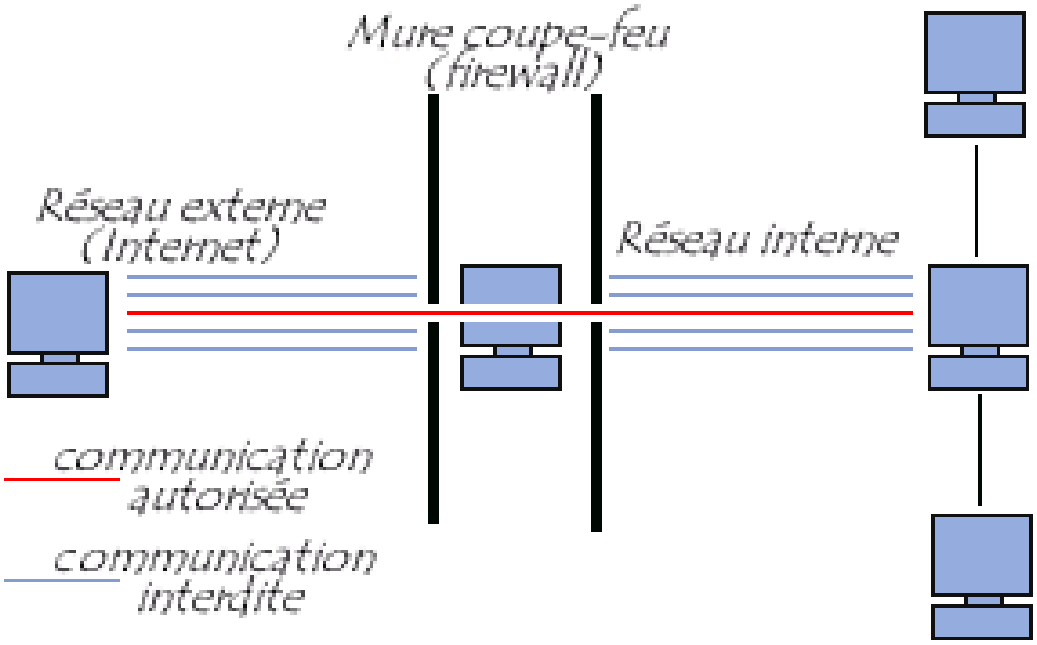
Des entités séparées sont crées, chacune responsable de zones de sécurité spécifiques du réseau d'entreprise.

B. Les FIREWALL : une technique de parade

1) Qu'est-ce qu'un pare-feu (Firewall)?

Un pare-feu (firewall en anglais), est un système physique (matériel) ou logique (logiciel) servant d'interface entre un ou plusieurs réseaux afin de contrôler et éventuellement bloquer la circulation des paquets de données, en analysant les informations contenues dans les couches 3, 4 et 7 du modèle OSI. Il s'agit donc d'une machine (machine spécifique dans le cas d'un firewall matériel ou d'un ordinateur sécurisé hébergeant une application particulière de pare-feu) comportant au minimum deux interfaces réseau :

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

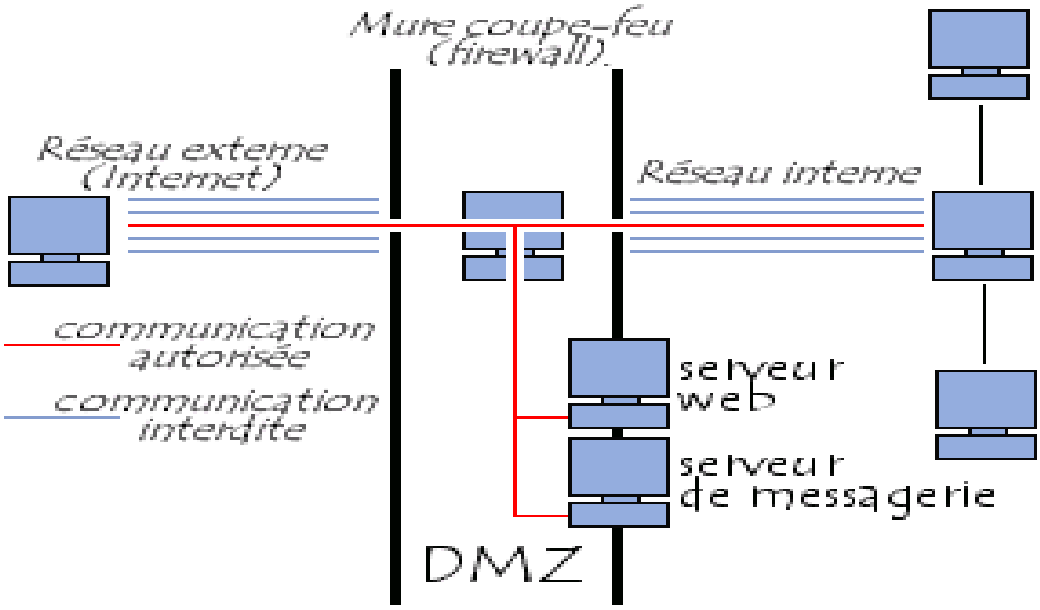


Le pare-feu représente ainsi généralement dans les entreprises un dispositif à l'entrée du réseau qui permet de protéger le réseau interne d'éventuelles intrusions en provenance des réseaux externes (souvent Internet). Les terminologies suivantes sont parfois également utilisées :

- garde-barriére (gate-keeper)
- porte coupe-feu
- antéserveur
- écluse

Lorsque certaines machines du réseau interne ont besoin d'être accessible de l'extérieur (comme c'est le cas par exemple pour un serveur web, un serveur de messagerie, un serveur FTP public, ...) il est souvent nécessaire de créer une nouvelle interface vers un réseau à part, accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'entreprise. On parle ainsi de **zone démilitarisé** (souvent notée

DMZ pour *DeMilitarized Zone*) pour désigner cette zone isolée hébergeant des applications mises à disposition du public.



Dans le cas où la zone protégée se limite à l'ordinateur sur lequel le firewall est installé on parle de *firewall personnel*.

2) - Le fonctionnement d'un système Firewall

Un système Firewall contient un ensemble de règles prédéfinies permettant :

- Soit d'autoriser uniquement les communications ayant été explicitement autorisées :

  "Tout ce qui n'est pas explicitement autorisé est interdit".

- 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 oeuvre 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 terme de communication.

3) Le filtrage de paquets

Le fonctionnement des systèmes pare-feu, historiquement assuré par les routeurs, est basé sur le principe du filtrage de paquets IP, c'est-à-dire sur l'analyse des en-têtes des paquets IP (aussi appelés *datagrammes*) échangés entre deux machines. En effet les machines d'un réseau relié à Internet sont repérées par une adresse appelée adresse IP.

Ainsi, lorsqu'une machine de l'extérieur se connecte à une machine du réseau local, et vice-versa, les paquets de données passant par le firewall contiennent les en-têtes suivants, qui sont analysés par le firewall:

- L'adresse IP de la machine émettrice
- L'adresse IP de la machine réceptrice
- Le type de paquet (TCP, UDP, ...)
- Le numéro de port (rappel: un port est un numéro associé à un service ou une application réseau)

Les adresses IP contenues dans les paquets permettent d'identifier la machine émettrice et la machine cible, tandis que le type de paquet et le numéro de port donnent une indication sur le type de service utilisé. Lorsque le filtrage est basé sur les adresses IP on parle de **filtrage par adresse** (*adress filtering*), tandis que le terme de **filtrage par protocole** (*protocol filtering*) est utilisé lorsque le type de paquets et le port sont analysés.

Certains ports sont associés à des service courants (les ports 25 et 110 sont généralement associés au courrier électronique, et le port 80 au Web) et ne sont généralement pas bloqués. Toutefois, il est recommandé de bloquer tous les ports qui ne sont pas indispensables (selon la politique de sécurité retenue).

Le port 23 par exemple est critique car il correspond au service Telnet qui permet d'émuler un accès par terminal à une machine du réseau de manière à pouvoir exécuter des commandes saisies au clavier à distance...

4) Le filtrage dynamique

Le fonctionnement décrit ci-dessus ne s'attache qu'à examiner les paquets IP, ce qui correspond au niveau 3 du modèle OSI. Or, de nombreux services (le FTP par exemple) initient une connexion sur un port statique, mais ouvrent dynamiquement (c'est-à-dire de manière aléatoire) un port afin d'établir une session entre la machine faisant office de serveur et la machine cliente. Ainsi, il est impossible de prévoir les ports à laisser passer ou à interdire. Pour y remédier, l'entreprise *Check point* a breveté un système de **filtrage dynamique de paquets** (le terme anglo-saxon exact étant *stateful inspection*) basé sur l'inspection des couches 3 et 4 du modèle OSI, permettant d'effectuer un suivi des transactions entre le client et le serveur et donc d'assurer la bonne circulation des données de la session en cours.

Si le filtrage dynamique est plus performant que le filtrage de paquets basique, il ne protége pas pour autant de failles applicatives, c'est-à-dire les failles liées aux logiciels, représentant la part la plus importante des risques en terme de sécurité.

5) - Le filtrage applicatif

Le filtrage applicatif permet, comme son nom l'indique, de filtrer les communications application par application, ce qui signifie qu'il travaille au niveau de la couche 7 du modèle OSI. Le filtrage applicatif suppose donc une connaissance de l'application, et notamment de la manière de laquelle elle structure les données échangées.

Un firewall effectuant un filtrage applicatif est appelé *passerelle applicative* car il permet de relayer des informations entre deux réseaux en effectuant un filtrage fin au niveau du contenu des paquets échangés. Il s'agit donc 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.

C) –Outil du travail : IPTables

1) Qu'est-ce que c'est ?

C'est l'interface de configuration de Netfilter. Netfilter c'est l'ensemble des fonctions internes au noyau (version 2.4 et supérieur) qui réalisent les opérations de firewalling

2) Que pouvons-nous faire avec iptables?

* monter un firewall filtrant basé sur les paquets mais aussi sur le statut des connexions engendrés par les paquets(le suivi de connexion)

* utiliser NAT(Network Address Translation) et le masquerading afin de partager un accés Internet à plusieurs machines

* utiliser NAT pour faire du proxy transparent (évite d'avoir à paramétrer le proxy sur les Clients/navigateurs web)

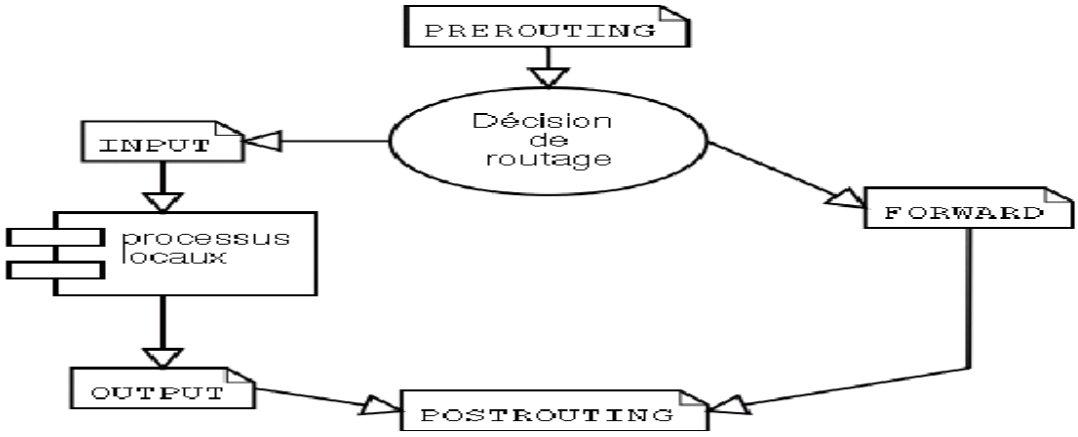
* mettre en place (notamment en permettant le marquage des paquets via la table mangle) la possibilité d'utiliser iproute2 dans le but d'obtenir un routeur sophistiqué permettant le QoS (Quality of Service, privilégié certains services, mettre en place des limites d'utilisation de bande passante sur un utilisateur, sur un groupe…)

* manipuler des paquets pour par exemple altéré le champ TOS(2) d'un datagramme IP

3) C'est quoi une chaîne ?

Nous allons donc nous positionner comme étant sur la machine qui fera office de firewall/routeur pour tenter de la sécuriser.

Lorsqu'un paquet arrive, il va être orienté (selon un certain nombre de paramètres) dans l'une des différentes chaînes disponibles.



Ainsi que nous le montres le schéma un paquet rentrera toujours dans la machine via la chaîne PREROUTING et sortira toujours de la machine via la chaîne POSTROUTING(chaînes servant notamment à certaines opérations de routage entre les 2 réseaux) raccordés par notre routeur.

Les chaînes INPUT et OUTPUT quand à elle serviront respectivement à placer des règles pour les paquets destinés à la machine et ceux émis par la machine, pour faire simple : si un paquet est destiné à ma machine, « il arrivera dans la chaîne INPUT » .

Par exemple si je demande la visualisation d'une page sur le web depuis la machine, j'émets une requête qui sortira par la chaîne OUTPUT et la réponse arrivera sur ma machine par la chaîne INPUT .

La nuance est dans le "destiné", à savoir que l'on peut considérer qu'un paquet à destination du LAN sera, au moins à certain(s) moment(s) et à certain(s) niveau(x) du modèle OSI, destiné à la machine faisant office de routeur mais sera lui orienté dans la chaîne FORWARD et non pas dans la chaîne INPUT.

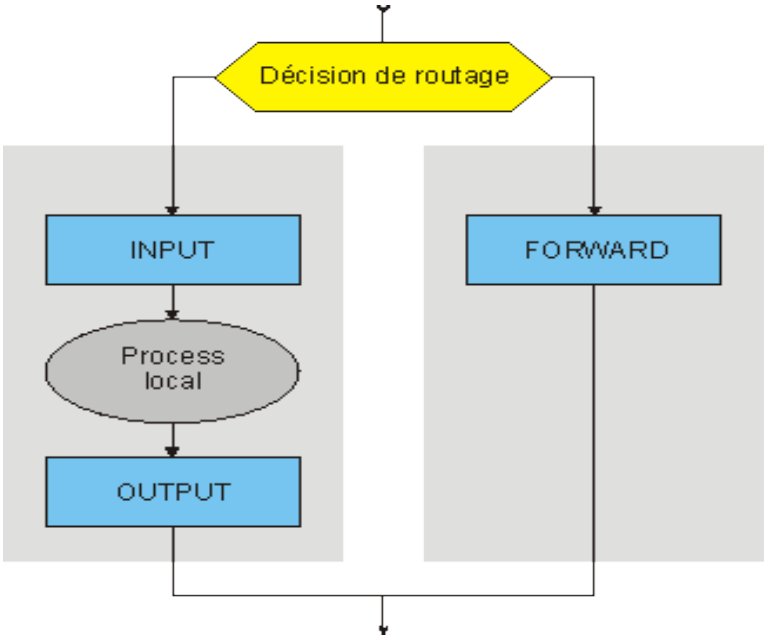
Au moment où le paquet rentre dans la chaîne, les règles correspondant à cette chaîne sont appliquées dans l'ordre dans lequel elles sont stockées.

4) La table de filtrage : Filter

C'est la table qui va permettre de filtrer tous les paquets qui entrent et sortent de notre machine. Il n'y a ici aucune modification de ces paquets, ils seront comparés à des critères définis dans la table Filter. Dans notre cas, il peut se passer deux choses différentes:

Un paquet qui entre est destiné à un processus de l'hôte (serveur HTTP, FTP...).

Un paquet qui entre est destiné à un autre réseau, c'est alors une fonction de routage.



Un paquet entre dans notre machine. Peu importe par quelle interface il entre, il peut venir aussi bien du réseau local que de l'Internet. Il passe d'abord par la fonction de décision de routage. C'est elle qui va déterminer si le paquet est destiné à un processus local de l'hôte ou à un hôte sur un autre réseau.

- Si le paquet est destiné à l'hôte local:
  - Il traverse la chaîne INPUT
  - S'il n'est pas rejeté, il est transmis au processus impliqué. Ce processus va donc le traiter et éventuellement émettre un nouveau paquet en réponse.

    - Ce nouveau paquet traverse la chaîne OUTPUT
    - S'il n'est pas rejeté, il va vers la sortie.
- Si le paquet est destiné à un hôte d'un autre réseau:
    - Il traverse la chaîne FORWARD.
    - S'il n'est pas rejeté, il poursuit alors sa route.

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

Pour faire cela on doit ajouter des règles de filtrage dans la table filter, on doit d'abord spécifié la table filter par –t FILTER puis , la chaîne sur laquelle on va filtrer par –A « nom chaîne » et enfin d'autre option pour la sélection des paquets et d'autres pour les opérations à appliquer.

Quelques Exemples

Voici une suite de quelques exemples pouvant être utiles. Ils seront accompagnés d'une explication sur leurs rôles.

```
> iptables -t filter -A INPUT --source 207.46.134.190 --jump DROP
```

Cet exemple permet de jeter tout ce qui vient de l'adresse 207.46.134.190. La spécification de la table filter est facultative car par défaut c'est celle qui est utilisée. On pourrait abréger cette commande en la suivante :

```
> iptables -A INPUT -s 207.46.134.190 -j DROP
```

Qui produit exactement le même résultat. L'utilisation de -A ajoute cette règle à la fin de la chaîne. Donc si une précédente règle laissait passer des paquets, ils n'arriveront pas jusqu'à celle-ci.

Pour construire un ensemble de règles de filtrage, on commencera généralement par supprimer toutes celles créées précédemment.

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

Cette commande vide intégralement la table filter de toutes les règles qui auraient précédemment été définies.

Ensuite on indiquera plutôt en premier tout ce que l'on accepte, pour ensuite refuser tout le reste. On commencera donc par des règles ressemblant à la suivante :

```
> iptables -t filter -A INPUT --protocol tcp --destination-port 80 --jump ACCEPT
```

Cette règle permet de laisser passer tout le trafic TCP entrant sur le port 80. Ce sera utile par exemple si un serveur web (par défaut sur le port 80) est présent sur la machine. Mais il faut ensuite que les réponses envoyées puissent l'être. Dans ce cas, la règle précédente autorise des personnes extérieures à demander une page. Pour que le serveur puisse la transmettre, il faudra une règle comme suit :

```
> iptables -t filter -A OUTPUT --protocol tcp --source-port 80 --jump ACCEPT
```

Pour laisser les utilisateurs internes accéder aux sites Internet, on peut définir cette règle :

```
> iptables -t filter -A OUTPUT --protocol tcp --destination-port 80 --jump ACCEPT
```

Celle-ci permet donc de laisser passer les paquets à destination du port 80 d'une machine extérieure. Et pour que les utilisateurs puissent recevoir les pages envoyées par ces machines, il faudra ajouter :

```
> iptables -t filter -A INPUT --protocol tcp --source-port 80 --jump ACCEPT
```

Pour ne pas confondre ce que font les quatre règles précédentes, voici un résumé de ce qu'elles font :

- La première laisse entrer ce qui est pour le port 80 de la machine.

- La deuxième laisse sortir les paquets par le port 80 de la machine.
- La troisième laisse sortir ce qui est pour le port 80 d'une autre machine.
- La quatrième laisse entre ce qui vient du port 80 d'une autre machine.

Dans le cas où ce sont les seules choses permises sur cette machine, on terminera par indiquer que tout le reste doit être ignoré. Arriveront à ce point seulement les paquets qui n'auront pas répondu aux critères précédents. On peut donc alors les jeter (avec la cible DROP). Une règle toute simple pourrait être la suivante :

```
> iptables -t filter -A INPUT --jump DROP
> iptables -t filter -A OUTPUT --jump DROP
```

On a aussi REJECT qui peut être utilisé à la place de DROP lorsqu'on veut envoyer un message à l'émetteur pour le prévenir du refus de ces paquets.

5) La Table de translation d'adresse : NAT

La table NAT (Network Address Translation) est la table qui va nous permettre de faire la translation des adresses IP des paquets, aussi de faire la translation des ports ou un mélange des deux.

Le principe :

Dans un datagramme, en plus des données, on trouve également quelques informations concernant le protocole utilisé et des identificateurs de l'émetteur et du destinataire du datagramme. Ce sont ces identificateurs qui nous intéressent:

L'adresse IP du destinataire.

Le port du service utilisé sur le destinataire.

Ces informations constituent une "socket"(port), elles sont indispensables pour arriver à joindre le bon service sur le bon serveur, par exemple le service HTTP du serveur www.lr-ensa.ma.

L'adresse IP de l'émetteur.

Le port de réponse.

Ces informations constituent une autre "socket", elles sont indispensables pour que l'émetteur d'un paquet puisse espérer recevoir une réponse.

Avec les fonctions NATage, lorsqu'un paquet transite par notre passerelle, nous allons pouvoir manipuler ces sockets comme on veut. Par exemple, nous pourrons changer l'adresse de l'émetteur ou le port de l'émetteur ou les deux. Nous pouvons aussi changer l'adresse du destinataire, ou le port du destinataire, ou les deux.

A quoi sert la NAT ?

La NAT nous permet de faire une multitude de choses dont les plus intéressantes:

Le masquage d'adresse

C'est une fonction fondamentale lorsque l'on souhaite connecter un réseau privé à l'Internet lorsque l'on ne dispose que d'une seule IP valide sur le Net, même si celle-ci est dynamique, ce qui est le cas qui nous intéresse le plus (Les clients sont sur le réseau privé et les serveurs sont sur le Net). C'est une forme particulière de **SNAT** (Source NAT)
C'est ce que sont capables de faire tous les routeurs SOHO (Small Office, Home Office) qui permettent de relier un petit réseau local à l'Internet, lorsque l'on ne dispose que d'un accès RTC, NUMERIS, Câble, ADSL... Un simple PC équipé d'un Linux 2.4.x permet de le faire aussi bien sinon mieux.

Le NAT de destination

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

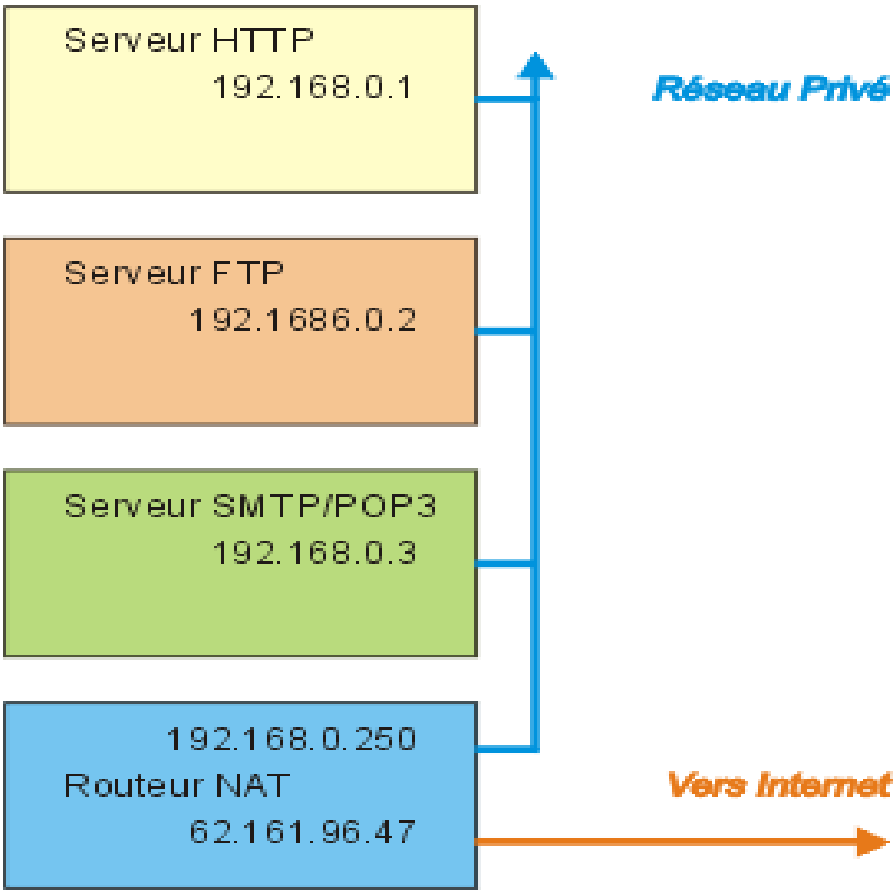
Comment faire ? La plus simple des solutions consiste à placer tous ces serveurs sur la même machine, celle qui a la seule adresse IP valide sur le Net à laquelle j'ai droit."

Mais comment assurer un minimum de sécurité sur une machine ouverte de tous les côtés ?

Comment faire pour assurer une disponibilité suffisante à chaque service dans les montées en charge ?

Cette solution ne parait finalement pas très acceptable, mais comment faire autrement? Tout simplement avec NAT. La machine frontale sera un simple routeur NAT. Côté Internet, elle possède la seule IP valide disponible, elle va faire croire que tous les services sont dessus, mais en réalité, lorsqu'elle va recevoir un paquet dont le socket de destination est

62.161.96.47:80, elle va remplacer ce socket par 192.168.0.1:80 et router le paquet vers le serveur HTTP. Lorsque la réponse du serveur va lui parvenir, elle remplacera le socket de l'émetteur (192.168.0.1:80) par 62.161.96.47:80 et enverra ça sur le Net.

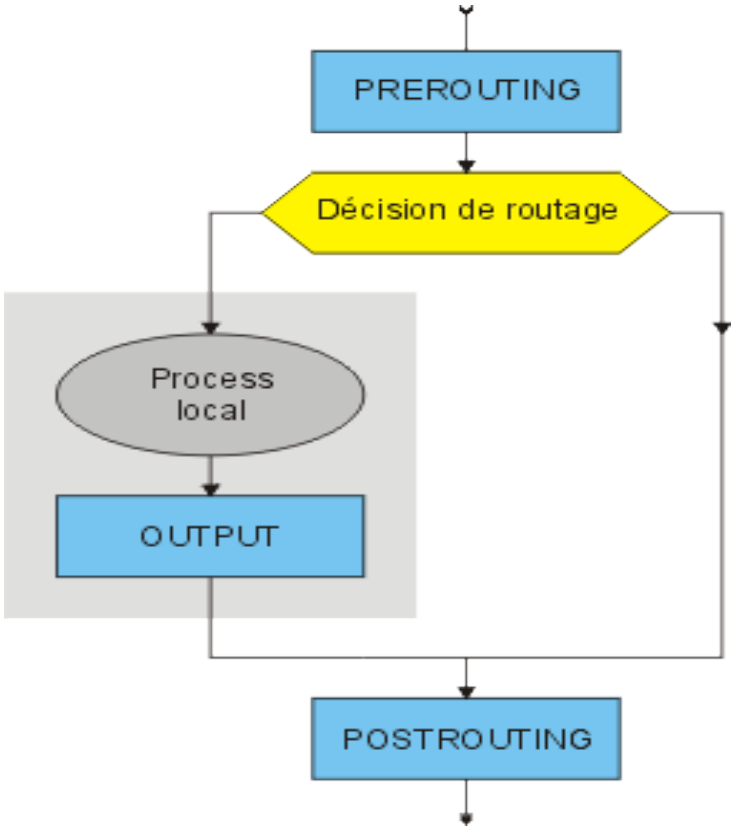


Bien entendu, le routeur NAT est capable de faire ça pour chacun des autres serveurs: Ce qui lui arrivera sur les ports 20 et 21 sera redirigé sur le serveur FTP (en réalité, le cas du FTP est bien plus difficile à résoudre que ça.

Ce qui lui arrivera sur le port 25 sera redirigé sur le serveur SMTP/POP3 service SMTP

Ce qui lui arrivera sur le port 110 sera redirigé sur le serveur SMTP/POP3 service POP3.

Le fonctionnement de NAT Et comment ça marche ?



La table NAT est organisée comme l'indique le schéma ci-dessus:

La chaîne PREROUTING va exécuter un traitement sur les "sockets" avant les décisions de routage. Nous nous en servirons pour faire du DNAT (Destination NAT), autrement dit, pour modifier la "socket" du destinataire.

La chaîne POSTROUTING intervient à la sortie du routeur. Elle servira à faire du SNAT (Source NAT) dont par exemple, le masquage d'adresse.

La chaîne OUTPUT, quant à elle, permet de modifier le socket de destination d'un paquet issu d'un processus local. « L'utilité de cette chaîne n'est pas évidente, dans la mesure

où, normalement, les paquets sortant d'un processus local devraient aussi passer par POSTROUTING ».

La seule possibilité supplémentaire est de pouvoir rediriger les paquets qui sortent d'un processus local à destination d'une cible extérieure, vers un autre processus local (127.0.0.1). Les possibilités offertes par le NAT sont quasiment infinies. Nous avons vu les plus fréquentes :

Masquage d'adresse, pour permettre à tout un réseau privé d'accéder au Net lorsque l'on ne dispose que d'une seule adresse IP valide sur le Net, redirection d'un service serveur adressé sur la passerelle vers un serveur situé dans le réseau privé, ça peut être utile pour les joueurs en réseau, mais aussi pour des applications plus professionnelles.

Pour que tout ça fonctionne correctement, le système s'appuie sur le suivi de connexion. Nous pouvons donc nous attendre à trouver des modules spécialisés pour certains protocoles, dont le FTP. Ainsi, le module ip_nat_ftp sera nécessaire si vous voulez travailler proprement en FTP.

Quelques Exemples

Le DNAT ou NAT Destination

On substitue à l'adresse de destination des paquets provenant du réseau public, une adresse du réseau local privé. Dans l'exemple, les paquets à destination de la machine 195.x sont redirigés vers la machine 172.y. On ne tient pas compte du port.

Dans cet exemple nous allons d'abord effacer toutes les regles (-F) de la table part defaut (Filter) ensuite on accepte tous les paquets . Puis on efface(-F) aussi toutes les regle de la table NAT (-t nat) pour les paquets qui ne sont pas encore routés . Enfin on défini une nouvelle regle dans laquelle les paquets à destination de la machine 195.115.19.35 sont redirigés vers la machine 172.16.0.1.

```
iptables -F INPUT  ; iptables -P INPUT ACCEPT

iptables -F OUTPUT  ; iptables -P OUTPUT ACCEPT
```

```
iptables -F FORWARD ; iptables -P FORWARD ACCEPT

iptables -t nat -F PREROUTING

iptables -t nat -A PREROUTING -d  195.115.19.35/32 \
         -j  DNAT --to-destination  172.16.0.1/32
```

Le SNAT ou NAT Source

Le SNAT consiste à substituer une adresse source dans un paquet sortant à son adresse source d'origine. On substitue ici, aux requêtes provenant du réseau 192.168.0.0/24, une des 10 adresses publiques.

Dans cet exemples

```
iptables -t nat -F POSTROUTING

iptables -t nat -A POSTROUTING -s 192.168.0.0/24 \
         -j  SNAT --to-source   195.115.90.1-195.115.90.10
```

L'IP Masquerade

Dans ce cas, les adresses privées, utilisent toutes la même adresse publique. C'est le procédé qui est utilisé avec ipchains. Il s'agit en fait de translation de port avec ipchains ou de SNAT (Nat Source) avec iptables.

```
iptables -t nat -F POSTROUTING

iptables -t nat -A POSTROUTING -s 10.10.10.0/8 \
         -j  SNAT --to-source 139.63.83.120
```

Une autre option consiste à utiliser l'option "MASQUERADE"

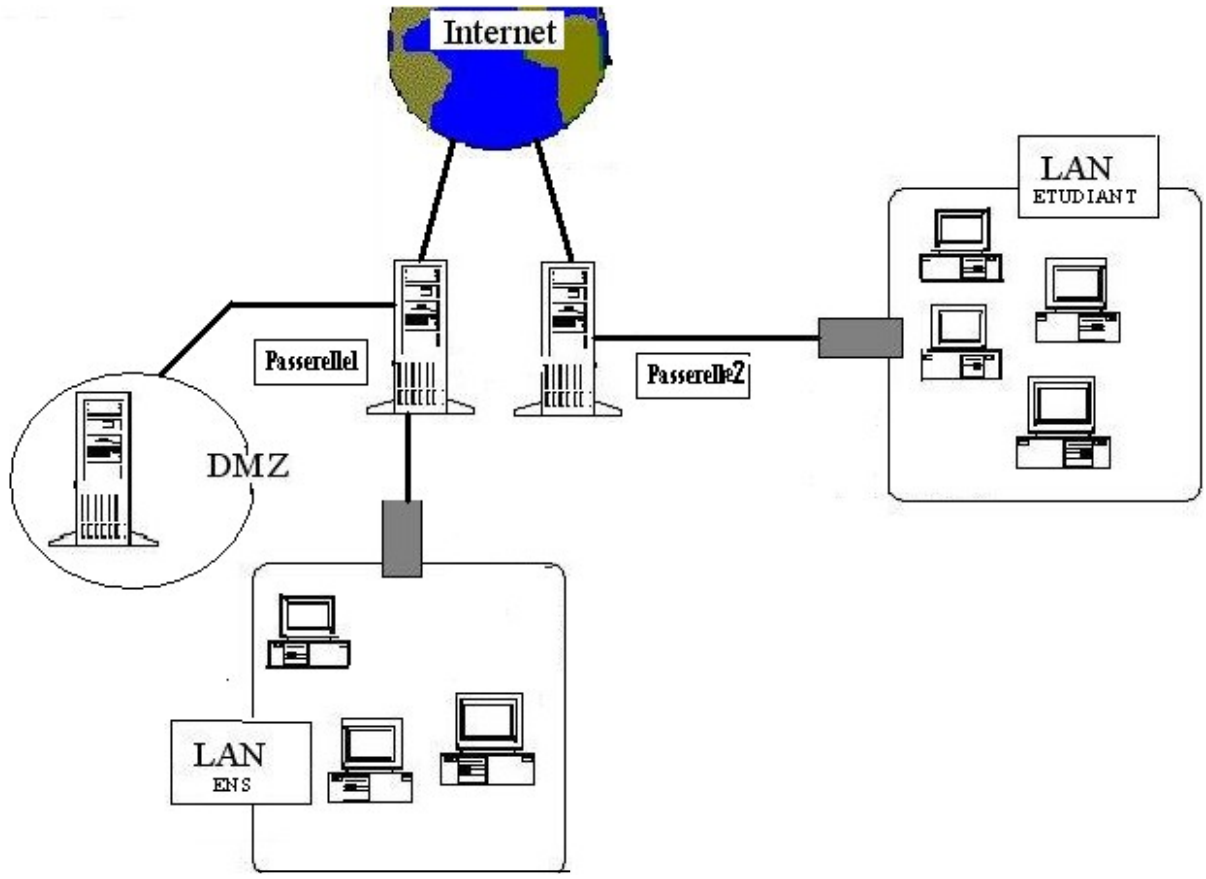
```
iptables -t nat -F POSTROUTING

iptables -t nat -A POSTROUTING -s 10.10.10.0/8 -j  MASQUERADE
```

D) Analyse de l'architecture réseaux de l'ESTO

1) Etude de l'existant

L'école supérieure de technologie dispose déjà d'une architecture réseaux avec un dispositif de sécurité basé sur le firewall.
L'architecture précédente était composée de quatre sous réseaux dont trois pour les salles de TP de chaque département et un pour l'administration et les enseignants. En plus de tout cela on avait une DMZ (zone démilitarisée) avec des adresses réelles.
L'architecture actuelle , version évoluée de la précédente, fait jumeler toute les salles de TP des étudiants donnant ainsi un seul sous réseaux «étudiant » au lieu de trois .Puis nous avons toujours le sous réseaux de l'administration et des enseignants . Et enfin la DMZ.
Ce qui nous donne le schéma suivant :



2)Critique de l’existant

Nous avons ici deux sous réseaux, celui des étudiants et celui des enseignants et l’administration, qui sont connectés sur Internet indépendamment l’un de l’autre par deux passerelles différentes. Ces sous réseaux ne communiquent pas entre elles. On ne peut « pinguer » une machine du réseaux des étudiants à partir du réseaux des enseignant et inversement. D’où nous constatons déjà un problème. Deux sous réseaux d’un même réseau qui ne communique pas. Encore pour accéder au serveur dans la DMZ on est obligé de passer par le net. Aussi le fait d’utiliser deux passerelles différentes n’est pas théoriquement un problème mais pourrait bien être mieux autrement, aussi on constate qu’il y a plusieurs adresses réelle utiliser dans cette architecture.

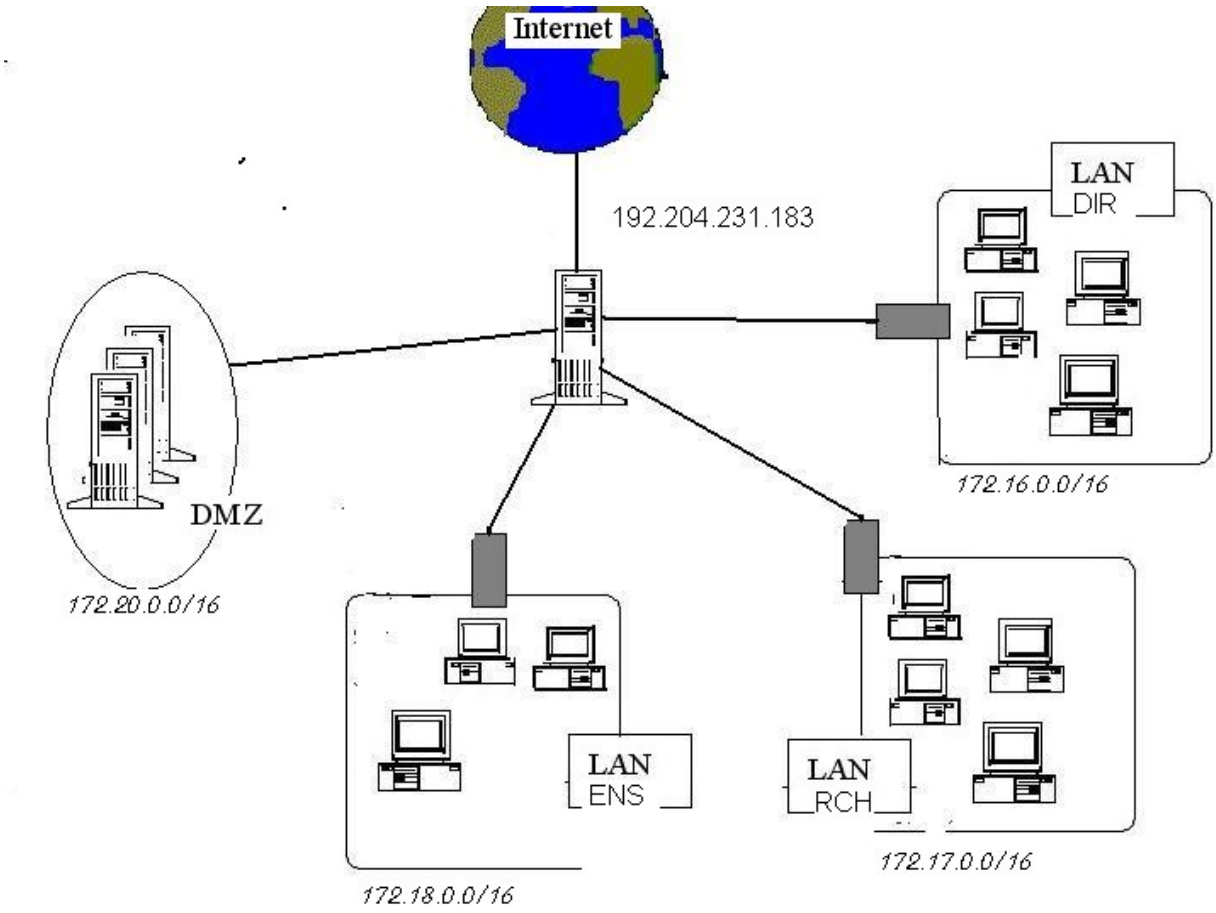
E)Travail effectué

1) Buts escomptés

En vu de trouver une résolution adéquate aux différents problèmes trouvés dans l'architecture actuelle nous avons élaboré une nouvelle architecture dans le but de :

- permettre la possibilité de communication entre les différents sous-résaux
- une bonne gestion du matériel et des adresses réelles
- gérer les priorités entre les différents sous réseaux en déterminant les droits d'accès de chaque sous réseaux.
- Eviter les pénétrations de l'extérieur(Internet) à notre réseau tout en gardant l'accès à Internet.

2) Schéma de la nouvelle architecture



3)descriptions du schéma

on a quatre sous réseaux

- DMZ : contient les serveurs : WEB,FTP,MAIL (172.20.0.0/16)
- DIR : regroupe les machines de la direction(172.16.0.0/16)
- CHR : regroupe les machines des enseignants(172.17.0.0/16)
- ENS : regroupes les machines des salles de TP(172.18.0.0/16)

et une seule passerelle (192.204.231.183)

4)Définition des droits d'accès :

-toutes les machines ont le droit d'accéder à l'INTERNET

- Les machines des sous-réseaux DIR,CHR,ENS ont le droit d'accéder à DMZ
- Les machines de DIR ont le droit d'accéder aux CHR et ENS, mais l'accès dans le sens inverse est interdit.
- Les machines de CHR ont le droit d'accéder à ENS, mais l'accès dans le sens inverse est interdit.
- L'accès au réseau interne de l'EST depuis l'extérieur est interdit sauf à DMZ.

5) Script de réalisation

```
IPTABLES=/sbin/iptables
# les variables qui rendra notre fichier de configuration paramétrable
EXTIF="eth0"
INTIF1="eth1"
INTIF2="eth2"
INTIF3="eth3"
INTIF4="eth4"
DIR="172.16.0.0/16"
RCH="172.17.0.0/16"
ENS="172.18.0.0/16"
DMZ="172.20.0.0/16"
SER_MAIL="172.20.0.3"
SER_FTP="172.20.0.2"
SER_WEB="172.20.0.1"
NAT="194.204.231.185"
echo "   External Interface:   $EXTIF"
echo "   Internal Interface1 DIR:  $INTIF1"
echo "   Internal Interface2 RCH :  $INTIF2"
echo "   Internal Interface3 ENS:  $INTIF3"
echo "   Internal Interface4 DMZ:  $INTIF4"




#activation du forwarding qui joue le rôle du routage
echo "1" > /proc/sys/net/ipv4/ip_forward

#bloquer tout le trafic dans notre réseau
$IPTABLES -P INPUT DROP
$IPTABLES -F INPUT
$IPTABLES -P OUTPUT DROP
$IPTABLES -F OUTPUT
$IPTABLES -P FORWARD DROP
$IPTABLES -F FORWARD
$IPTABLES -t nat -F

# l'accès à l'Internet

$IPTABLES -A FORWARD -s $DIR -i$INTIF1 -o $EXTIF ACCEPT
$IPTABLES -A FORWARD -s $RCH -i$INTIF2 -o $EXTIF ACCEPT
$IPTABLES -A FORWARD -s $ENS -i$INTIF3 -o $EXTIF ACCEPT
```

```
$IPTABLES -A FORWARD -s $DMZ -i$INTIF4 -o $EXTIF ACCEPT

# Enseignants ont l'accès aux poste des étudiants

$IPTABLES -A FORWARD -s $RCH -i$INTIF2 -o $INTIF3 ACCEPT
$IPTABLES -A FORWARD -s $ENS -i$INTIF3 -o $INTIF2 -j ACCEPT -m
state --state ESTABLISHED,RELATED

# Direction a le droit d'accès aux postes des étudiants et aux postes des
enseignants

$IPTABLES -A FORWARD -s $DIR -i$INTIF1-o $INTIF2 ACCEPT
$IPTABLES -A FORWARD -s $DIR -i$INTIF1-o $INTIF3 ACCEPT
$IPTABLES -A FORWARD -s $ENS -i$INTIF3 -o $INTIF1 -j ACCEPT -m
state --state ESTABLISHED,RELATED
$IPTABLES -A FORWARD -s $RCH-i$INTIF2 -o $INTIF1 -j ACCEPT -m
state --state ESTABLISHED,RELATED

# Accès à DMZ localement

$IPTABLES -A FORWARD -s $DIR -i$INTIF1 -o $INTIF4 -j ACCEPT
$IPTABLES -A FORWARD -s $RCH -i$INTIF2 -o $INTIF4 -j ACCEPT
$IPTABLES -A FORWARD -s $ENS -i$INTIF3 -o $INTIF4 -j ACCEPT
$IPTABLES -A FORWARD -s $DMZ -i$INTIF4 -o $INTIF1 -j ACCEPT -m
state --state ESTABLISHED,RELATED
$IPTABLES -A FORWARD -s $DMZ -i$INTIF4 -o $INTIF2 -j ACCEPT -m
state --state ESTABLISHED,RELATED
$IPTABLES -A FORWARD -s $DMZ -i$INTIF4 -o $INTIF3 -j ACCEPT -m
state --state ESTABLISHED,RELATED

# translation des adresses Snat

$IPTABLES -A POSTROUTING -t nat -o EXTIF -j SNAT --to $NAT

# accès au serveur web ftp mail depuis un poste local
```

$IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 80 -j DNAT --to /$SER_WEB:80

$IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 21 -j DNAT --to /$SER_FTP:21

$IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 20 -j DNAT --to /$SER_FTP:20

$IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 25 -j DNAT --to /$SER_FTP:25

$IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 110 -j DNAT --to /$SER_FTP:110

 # redirection des connections HTTP ,FTP,SMPT,POP...

```
   # on charge le module ip_contrack qui suit les connections ftp

      modprobe ip_conntrack_ftp
    $IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 80 -j DNAT
--/to-destination $SER_WEB:80
    $IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 20 -j DNAT
--/to-destination $SER_FTP:20
    $IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 21 -j DNAT
--/to-destination $SER_FTP:21
    $IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 25 -j DNAT
--/to-destination $SER_MAIL:25
    $IPTABLES -t nat -A PREROUTING -d $NAT -p tcp --dport 110 -j DNAT
-/-to-destination $SER_MAIL:110
```

 # accepter les connections aux serveurs depuis l'Internet

```
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF4 -m multiport --dport /
80,21,20,25,110 -j ACCEPT -m state --state ESTABLISHED,RELATED,NEW

$IPTABLES -A FORWARD -o $EXTIF -i $INTIF4 -j ACCEPT -m state --state
/ESTABLISHED,RELATED,NEW
```