



Réseaux - Firewalling - DMZ

Conception d'une Zone Démilitarisée (DMZ)

Abderrahim ESSAIDI

Vivien BOISTUAUD

Ngoné DIOP

Université de Marne la vallée - UFR Ingénieurs 2000
Informatique et Réseaux – 2^{ème} année
Année Universitaire 2006-2007

Table des matières

| | |
|---|----|
| Table des matières | 2 |
| Introduction | 3 |
| 1 Plan d'adressage du réseau | 4 |
| 2 Mise en place d'une stratégie de routage classique | 5 |
| 2.1 Configuration des interfaces réseaux | 5 |
| 2.2 Détection des services fournis par le routeur | 6 |
| 2.3 Configuration des passerelles par défaut | 7 |
| 3 Configuration des services de la zone DMZ | 8 |
| 3.1 Configuration d'un serveur Web supportant le PHP | 8 |
| 3.1.1 Installation | 8 |
| 3.1.2 Informations sur les spécificités Debian d'Apache 2 | 9 |
| 3.1.3 Configuration | 10 |
| 3.1.4 Le processus httpd | 10 |
| 3.1.5 Test de la configuration | 12 |
| 3.1.6 Activation du module PHP | 12 |
| 3.2 Configuration d'un serveur DHCP | 14 |
| 3.2.1 Introduction au protocole DHCP | 14 |
| 3.2.2 Installation du serveur DHCP sur la machine DMZ | 15 |
| 3.2.3 Installation d'un relai DHCP sur la machine routeur | 18 |
| 3.2.4 Configuration du client DHCP sur le PC LAN | 19 |
| 3.2.5 Tests de la configuration : Assignation et renouvellement | 21 |
| 3.3 Test de la sécurité pour le PC de la DMZ | 23 |
| 4 Sécurisation de la plateforme | 24 |
| 4.1 Introduction au firewall statefull iptables | 24 |
| 4.2 Sécurisation des communications par restriction | 25 |
| 4.3 Filtrage interne | 27 |
| 4.3.1 Support des paquets ICMP entre LAN, routeur et DMZ | 27 |
| 4.3.2 Support du relai DHCP | 28 |
| 4.3.3 Support des autres services de la DMZ | 30 |
| 4.4 Translation d'adresses (NAT) | 31 |
| 4.5 Translation de ports (PAT) | 32 |
| 4.5.1 Mise en place d'une translation de port | 32 |
| 4.5.2 Tests de fonctionnement de la translation de ports | 33 |
| Conclusion | 35 |

Introduction

L'objectif de ce TP est de concevoir une architecture de réseau d'entreprise comprenant une zone démilitarisée (DMZ), un réseau local et un accès à internet. Ces réseaux seront reliés ensemble par un serveur linux assurant le routage des paquets.

La zone démilitarisée hébergera des serveurs proposant des services au réseau local ou à internet, comme un serveur web (protocole http) ou un serveur DHCP. Le but est ainsi d'isoler les postes clients (du réseau LAN) de la zone de fourniture de services (DMZ) et de les protéger contre les attaques provenant des réseaux extérieurs (internet).

Pour effectuer ce TP, nous avons formé un trinôme. Notre but était de simuler un réseau réel d'entreprise en utilisant un premier PC représentant le réseau LAN, un second PC qui représente la zone démilitarisée comportant un serveur http et dhcp, et un troisième PC prenant en charge la fonction de routage.

Une fois notre réseau configuré, nous avons ensuite interconnecté notre « routeur » avec le routeur du trinôme voisin pour élargir le réseau et simuler des accès extérieurs. La connexion entre notre routeur et le routeur du réseau voisin sera assimilée, dans la suite de ce rapport, à une connexion internet.

1 Plan d'adressage du réseau

Pour identifier les composants de notre réseau (LAN + R1 + DMZ), nous utilisons la plage d'adresse 194.10.x.x.

Le LAN a pour adresse de réseau 194.10.10.0/24 et le DMZ a pour adresse de réseau 194.10.20.0/24. Enfin, le réseau entre les deux routeurs a pour adresse 194.10.30.0/24.

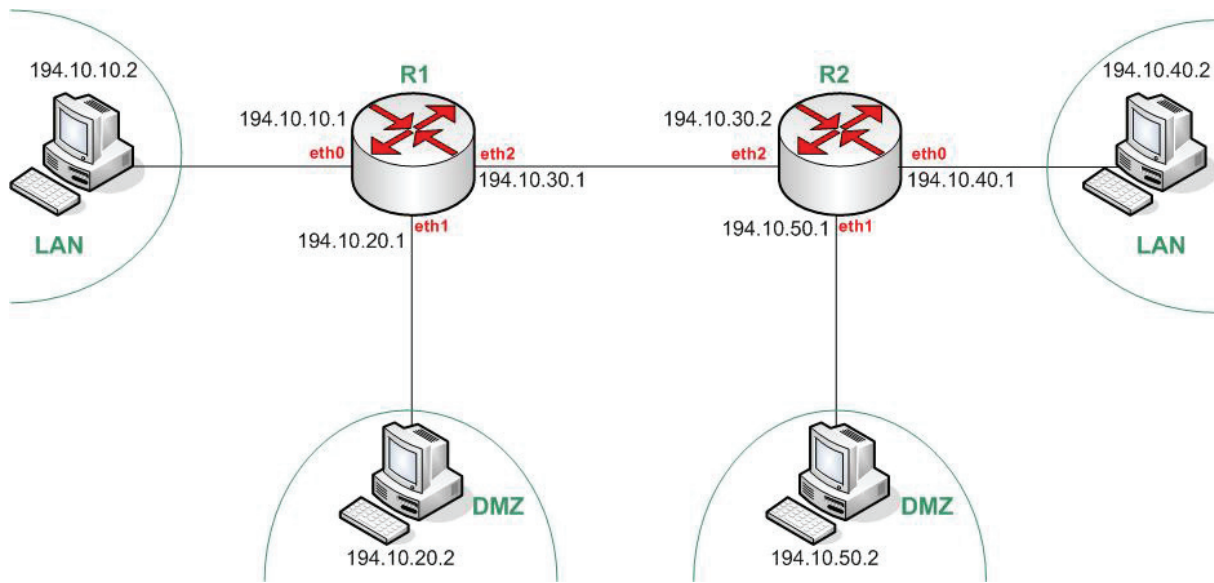


Figure 1 - Plan d'adressage du réseau d'entreprise et du réseau étranger

Les routeurs (R1 et R2) sont des PC constitués chacun de 3 cartes Ethernet :

- eth0 fait l'interface avec le réseau LAN
- eth1 fait l'interface avec le DMZ
- eth2 fait l'interface avec le routeur R2, qui fait partie d'un autre réseau (internet)

Chacune des autres machines est équipée d'une seule carte Ethernet, nommée eth0. Les câbles réseaux utilisés pour interconnecter les machines sont des câbles croisés, ce qui est indispensable pour connecter deux machines sans utiliser de matériel intermédiaire (comme un hub, un switch ou un routeur).

Tous les PC fonctionnent sous la version Sarge de Debian GNU/Linux, en utilisant un noyau linux de la famille 2.6. Les PC connectés au LAN peuvent utiliser potentiellement n'importe quel système d'exploitation prenant en charge un périphérique Ethernet et le protocole TCP/IP.

Cependant, les manipulations nécessaires sur ces systèmes (Windows, MacOS, MacOS X, Solaris, HP-UX) ne seront pas décrites dans la suite de ce rapport.

2 Mise en place d'une stratégie de routage classique

Dans cette partie, nous allons réaliser d'une part l'interconnexion des 3 PC puis d'autre part celle des deux « routeurs ». Pour cela, nous allons configurer leurs interfaces Ethernet, en activant le routage IP sur notre machine R1 et en configurant les tables de routage des PC sans appliquer de règles de filtrage particulière (pas de contrôle des accès).

2.1 Configuration des interfaces réseaux

Après avoir interconnecté les PC conformément à la Figure 1 ci-dessus, nous devons maintenant configurer les cartes réseaux de chaque PC en utilisant la commande `ifconfig`.

Sur le PC appartenant au réseau LAN, nous entrons la commande suivante :

```
# ifconfig eth0 194.10.10.2 netmask 255.255.255.0 up
```

Cette commande permet d'attribuer à la carte réseau `eth0` une adresse IP (conformément au plan d'adressage), un masque de réseau et enfin de l'activer.

Nous procédons de la même manière sur le PC appartenant à la zone démilitarisée en entrant la commande suivante :

```
# ifconfig eth0 194.10.20.2 netmask 255.255.255.0 up
```

Le PC qui aura pour fonction le routage interconnecte la DMZ, le LAN et le routeur R2. Il doit donc avoir trois interfaces, autrement dit dans notre cas trois cartes réseaux ethernet. Nous entrons alors les commandes suivantes pour les configurer :

```
# ifconfig eth0 194.10.10.1 netmask 255.255.255.0 up  
# ifconfig eth1 194.10.20.1 netmask 255.255.255.0 up  
# ifconfig eth2 194.10.30.1 netmask 255.255.255.0 up
```

Pour que ce PC puisse effectuer le routage, il faut activer cette fonctionnalité. Sous la distribution Debian, il faut modifier le fichier `/etc/network/options` et placer la valeur de l'option `ip_forward` à `yes`.

Cependant, en procédant comme cela, la modification ne sera prise en compte qu'au prochain démarrage. Si l'on veut que la prise en compte se fasse instantanément, il faut modifier directement le fichier virtuel du noyau `/proc/sys/net/ipv4/ip_forward` en tapant la commande :

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

2.2 Détection des services fournis par le routeur

Pour obtenir la liste des serveurs actifs sur la machine routeur nous utilisons la commande `netstat` comme suit :

```
# netstat -tan

Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale           Adresse distante         Etat
tcp      0      0 0.0.0.0:111              0.0.0.0:*                LISTEN
tcp      0      0 0.0.0.0:113              0.0.0.0:*                LISTEN
tcp      0      0 0.0.0.0:22               0.0.0.0:*                LISTEN
tcp      0      0 0.0.0.0:893              0.0.0.0:*                LISTEN
```

L'option `-t` permet d'afficher seulement les sockets utilisées par le protocole `tcp`.

L'option `-a` permet d'afficher à la fois les sockets qui sont à l'écoute mais aussi celles qui ne le sont pas, c'est-à-dire les connexions actuellement établies avec cette machine.

L'option `-n` permet d'afficher les adresses au format numérique au lieu d'essayer de déterminer le nom symbolique des hôtes, des ports ou des utilisateurs. Cela rend le fonctionnement de la commande plus rapide dans de nombreux cas et évite de solliciter des serveurs DNS.

On constate donc que les ports `tcp` qui sont à l'écoute (état `LISTEN`) sont les ports 111, 113, 22 et 893. Ces ports sont respectivement utilisés par les serveurs actifs SunRPC, identd (utilisé principalement en combinaison de l'IRC), Secured Shell (SSH). Le port 893 n'a pas été assigné par une autorité de certification.

L'option `-u` permet d'afficher les sockets utilisés par le protocole `udp`. Nous avons donc également utilisé la commande ci-dessous pour étudier les ports `udp` ouverts sur la machine routeur. Nous avons obtenu les informations suivantes :

```
# netstat -uan

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp      0      0 0.0.0.0:32768           0.0.0.0:*
udp      0      0 0.0.0.0:653            0.0.0.0:*
udp      0      0 0.0.0.0:656            0.0.0.0:*
udp      0      0 0.0.0.0:111            0.0.0.0:*
udp      0      0 0.0.0.0:32769           0.0.0.0:*
```

Les ports ouverts sont les ports 32768, 32769, 653, 656 et 111. Les ports 32768 et 32769 sont utilisés pour les partages NFS comme ports de contrôle de procédure distante (RPC – Remote Procedure Protocol). Les ports 653 et 656 sont utilisés par le client NFS. Le port 111 est, comme en `tcp`, le port SunRPC.

La colonne « `state` » n'affiche aucune information d'état car le protocole UDP est un protocole sans état (stateless protocol), contrairement au TCP.

2.3 Configuration des passerelles par défaut

Afin que le routage s'effectue correctement, il est nécessaire :

- Sur la machine routeur, d'entrer une route par défaut qui redirige sur la connexion internet (connexion avec le routeur R2).
- Sur les machines LAN et DMZ d'entrer les routes par défaut comme étant les adresses IP des interfaces du routeur connectées respectivement aux réseaux LAN et DMZ.

Pour configurer le PC routeur, nous utilisons la commande suivante :

```
# route add default gw 194.10.30.2
```

Cette commande permet de définir la route menant au routeur R2 comme étant la route par défaut.

De façon similaire, pour configurer le PC appartenant au LAN, nous entrons la commande suivante :

```
#route add default gw 194.10.10.1
```

Pour configurer le PC appartenant à la DMZ, nous entrons la commande suivante :

```
#route add default gw 194.10.20.1
```

Sur le PC routeur, nous obtenons la table de routage suivante :

| Destination | Passerelle | Genmask | Indic | Metric | Ref | Use | Iface |
|-------------|-------------|---------------|-------|--------|-----|-----|-------|
| 194.10.10.0 | * | 255.255.255.0 | U | 0 | 0 | 0 | eth0 |
| 194.10.30.0 | * | 255.255.255.0 | U | 0 | 0 | 0 | eth2 |
| 194.10.20.0 | * | 255.255.255.0 | U | 0 | 0 | 0 | eth1 |
| default | 194.10.30.2 | 0.0.0.0 | UG | 0 | 0 | 0 | eth2 |

On constate donc que le routeur connaît maintenant toutes les routes nécessaires pour atteindre n'importe quel segment de notre réseau et du réseau voisin via le routeur R2.

3 Configuration des services de la zone DMZ

Une DMZ (Demilitarized zone) est une zone tampon d'un réseau d'entreprise, située entre le réseau local et Internet, derrière le pare-feu. Il s'agit d'un réseau intermédiaire regroupant des serveurs publics (HTTP, DHCP, mails, DNS, etc.). Ces serveurs devront être accessibles depuis le réseau interne de l'entreprise et, pour certains, depuis les réseaux externes. Le but est ainsi d'éviter toute connexion directe au réseau interne.

Dans cette partie, nous allons configurer un serveur DMZ qui devra offrir deux services :

- un serveur HTTP sur lequel on peut héberger le site internet de l'entreprise
- un serveur DHCP qui va permettre d'assurer l'adressage des machines du réseau local

Nous avons configuré notre DMZ comme étant le réseau 194.10.20.0/24. L'interface du routeur se connectant sur ce réseau aura l'adresse IP 194.10.20.1. Les paquets en provenance de la zone DMZ seront envoyés vers cette interface. Comme décrit précédemment, nous avons configuré la machine hébergeant nos services pour qu'elle ait l'adresse IP 194.10.20.2 à l'aide des commandes :

```
# ifconfig eth0 194.10.20.2 netmask 255.255.255.0
```

Pour la configuration de l'interface, et

```
# route add default gw 194.10.20.1
```

Pour désigner l'interface du routeur connectée au réseau DMZ comme route par défaut.

3.1 Configuration d'un serveur Web supportant le PHP

Un serveur HTTP est un logiciel capable d'interpréter les requêtes HTTP qu'il reçoit et de fournir une réponse dans ce même protocole. Apache est le serveur HTTP le plus répandu sur Internet. Ce serveur est très extensible. Il permet en effet d'ajouter des modules supplémentaires qui enrichissent le serveur en termes de fonctionnalités. Dans cette partie, nous allons installer et configurer un serveur Apache et y ajouter un interpréteur de scripts PHP.

3.1.1 Installation

Sur la DMZ, nous disposons d'une machine fonctionnant sous le système d'exploitation Debian GNU/Linux. Cette distribution propose un outil de gestion de package, nommé `aptitude`, qui permet d'automatiser l'installation, la configuration et la mise à jour des logiciels. Pour installer apache 2 sous Debian, nous avons utilisé la commande suivante :


```
# apt-get install apache2
```

La version installée est la version 2.0, aujourd'hui la plus répandue. En effet, Apache 1.3 est une version plus ancienne, héritière du logiciel NCSA httpd, et Apache 2.2 est trop récent pour le moment (troisième release publique au moment de l'écriture de ce rapport).

3.1.2 Informations sur les spécificités Debian d'Apache 2

L'organisation des fichiers de configuration d'Apache 2 sous Debian GNU/Linux diffère légèrement de l'organisation classique de la distribution disponible sur le site web du projet apache (<http://httpd.apache.org>). Les fichiers sont stockés dans les répertoires `/etc/apache2`, `/etc/apache2/mods-available` et `/etc/apache2/sites-available`.

L'organisation des fichiers a été prévue de façon modulaire : les fichiers indispensables à la configuration d'apache sont dans le dossier `/etc/apache2`. Ainsi :

- `ports.conf` définit les adresses IP et ports d'écoute du serveur Apache 2.0 (directives `Listen`)
- `apache2.conf` définit la configuration générale du serveur, notamment la gestion des processus et des threads, les icônes, le répertoire principal de configuration, etc.

Le sous répertoire `mods-available/` contient les fichiers de chargement (`*.load`) et de configuration (`*.conf`) des modules apaches installés sur la machine.

Le sous répertoire `sites-available/` contient les fichiers de configuration des sites. Celui pour le site par défaut se nomme `default.conf`, les autres noms sont libres et permettent de définir des hôtes virtuels (Virtual Hosts).

Pour activer ou désactiver un module, il suffit de créer ou de supprimer un lien symbolique vers les fichiers `load` et `config` concernés dans le sous répertoire `mods-enabled/`. Une manipulation similaire est à faire avec les fichiers de définition d'hôtes dans le sous répertoire `sites-available/`.

Pour simplifier cette procédure, des scripts propres à Debian GNU/Linux sont installés avec Apache 2.0. `a2enmod nom_module` pour activer un module, `a2dismod nom_module` pour en désactiver un. De même, `a2ensite nom_site` permet d'activer un site, tandis que la commande `a2dissite nom_site` permet d'en désactiver un.

Ces spécificités de Debian peuvent être utilisées sur d'autres systèmes grâce aux directives suivantes du fichier `apache2.conf` :

```
ServerRoot "/etc/apache2"  
Include /etc/apache2/mods-enabled/*.load  
Include /etc/apache2/mods-enabled/*.conf
```

```
Include /etc/apache2/ports.conf
Include /etc/apache2/sites-enabled/[^.#]*
```

3.1.3 Configuration

Le protocole HTTP assigne le port 80 comme le porte d'échange des messages. La directive « `Listen` » du fichier de configuration de Apache `/etc/apache2/ports.conf` permet de définir les adresses IP et ports d'écoute du serveur. Nous modifions cette directive pour que notre serveur écoute sur le port 8080 de toutes ses interfaces de la manière suivante :

```
[...]
Listen 8080
[...]
```

La directive « `ServerName` » définit le nom d'hôte du serveur (nom de domaine ou adresse IP) sur lequel va fonctionner le serveur Apache. Nous le configurons pour être égal à l'adresse IP assignée à la machine DMZ.

```
[...]
ServerName 194.10.20.2
[...]
```

Le répertoire par défaut est spécifié par la directive « `DocumentRoot` ». Le `DocumentRoot` par défaut est dans le fichier de configuration « `/etc/apache2/site-available/default.conf` ». Il pointe sur « `/var/www/` » avec une redirection vers son sous-répertoire « `/apache2-default/` ».

La directive « `DirectoryIndex` » définit la liste des ressources à chercher lorsque le client requiert un index du répertoire par ajout du slash final à une URL pointant sur ce répertoire. Sur notre serveur, elle a la valeur `index.html`.

Ces deux dernières directives permettent de définir le fichier qui est envoyé par défaut lorsqu'on fait une requête HTTP sur l'adresse et le port du serveur Apache. C'est-à-dire pour notre cas, lorsqu'on tape l'adresse « `http://194.10.20.2:8080` » sur un navigateur Internet ou à l'aide d'une application utilisant le protocole http (comme l'application `wget`).

3.1.4 Le processus httpd

Lançons le serveur avec la commande suivante :

```
# apache2 -k start
```

Repérons-le dans la liste des processus. Pour cela, nous utilisons la commande :

```
# ps -aux | grep apache2
```

Une liste de cinq processus apache est affichée.

Dans le fichier de configuration « `/etc/apache2/apache2.conf` », on observe qu'il existe plusieurs modèles de gestion des threads et processus d'Apache 2. Les plus courants sous Linux sont les modèles `prefork` et `worker`.

Le module `worker.c` est principalement utilisé si le serveur apache doit faire des traitements externes plus lourds que le simple service de pages statiques, comme le service de pages PHP ou JSP (par interconnexion avec un conteneur de servlets/JSP comme Tomcat ou Jetty). Le module `prefork.c` est le module préconisé par défaut sous Linux pour le service de pages statiques.

Pour savoir quel est le module utilisé par notre instance d'apache, et qui a été choisit lors de la compilation des binaires de l'application, nous pouvons utiliser la commande suivante :

```
# apache2 -l
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  [...]
  worker.c
  http_core.c
  mod_mime.c
  mod_status.c
  [...]
```

Notre serveur utilise donc le module `worker.c` pour gérer son pool de connexions. On peut alors modifier le nombre de processus serveurs lancés au démarrage, le nombre de threads disponibles, ainsi que le nombre maximum de clients simultanés. Pour cela, on modifie la section « `<IfModule worker.c>` » ci-dessous en mettant par exemple la directive « `StartServers` » à 10.

```
[...]
<IfModule worker.c>
StartServers          10
MaxClients            150
MinSpareThreads       25
MaxSpareThreads       75
ThreadsPerChild       25
MaxRequestsPerChild   0
</IfModule>
[...]
```

Un redémarrage du serveur puis une nouvelle consultation de la liste des processus apache nous montre qu'il y'en a effectivement 10 qui ont été lancés.

3.1.5 Test de la configuration

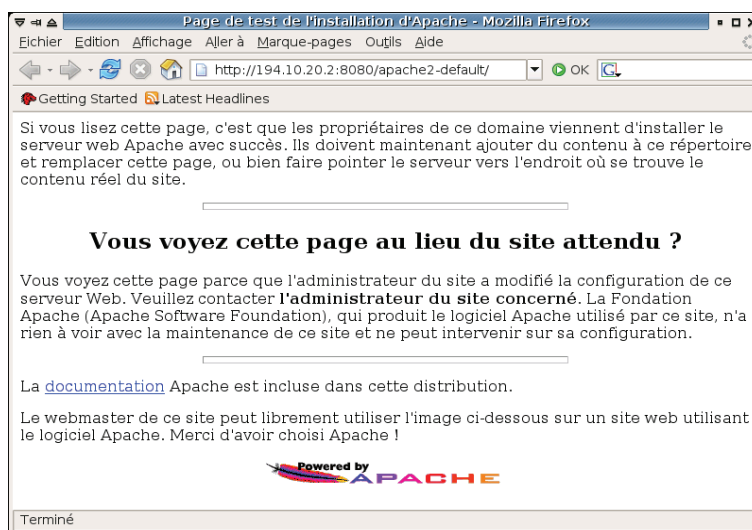


Figure 2 – Test de la configuration par défaut du serveur apache depuis le pc LAN

Comme, nous avons configuré notre serveur pour qu'il écoute et envoie ses réponses sur le port 8080, une requête HTTP 1.1 a été faite sur le port 8080 par l'intermédiaire du navigateur Internet Mozilla Firefox depuis le PC LAN. Le serveur a répondu avec une réponse HTTP 1.1 en redirigeant le navigateur sur l'adresse `http://194.10.20.2:8080/apache2-default/`, qui a renvoyée la page d'accueil par défaut (ici, le fichier `/var/www/apache2-default/index.html`).

Les mêmes tests, effectués depuis le réseau LAN 194.10.40.0/24 ont permis d'obtenir la même page. En effet, pour le moment, aucun filtrage n'est effectué sur les réseaux et l'ensemble des services des machines sont librement accessibles depuis les autres machines, même extérieures.

3.1.6 Activation du module PHP

PHP (Hypertext Preprocessor) est un langage de scripts libre principalement utilisé pour être exécuté par un serveur HTTP. Le serveur Apache est conçu pour supporter de nombreux modules lui donnant ainsi des fonctionnalités supplémentaires. Pour la plupart une simple activation du module est nécessaire. Pour PHP, la procédure est la suivante:

On commence par installer l'interpréteur PHP 4 et le module Apache 2 associé par l'intermédiaire de l'outil « `aptitude` » :

```
# apt-get install libapache2-php4
```

Cette commande va se charger d'effectuer automatiquement toute la configuration nécessaire dans les fichiers `/etc/apache2/mods-available/php4.conf` et `/etc/apache2/mods-available/php4.load`.

Il faudra néanmoins vérifier que la ligne suivante a été activée dans le fichier de configuration `mods-available/php4.conf`. Cette ligne fait appel à l'interpréteur PHP quand il doit servir un fichier ayant pour extension `php`, `phtml`, `php3` ou `php4`.

```
[...]  
AddType application/x-httpd-php .php .phtml .php3  
[...]
```

Il faut également vérifier que des liens symboliques vers `php4.load` et `php4.conf` ont été créés dans le dossier `/etc/apache2/mods-enabled`. Si ce n'est pas le cas, il faut utiliser la commande `a2enmod php4`, exécutée en tant que `root`.

Le module PHP activé, il faut maintenant activer le module MySQL de PHP4 qui va permettre aux scripts PHP le nécessitant de se connecter à une base de données de ce type.

Pour activer ce module, il faut éditer le fichier de configuration de PHP se trouvant dans « `/etc/php4/apache2/php.ini` ». Il suffit de supprimer le caractère # (commentaire) de la ligne suivante pour que PHP/MySQL soit prêt à être utilisé sur notre serveur.

```
[...]  
# extension=mysql.so  
[...]
```

Pour que cette modification soit prise en compte, il faut recharger la configuration du serveur Apache 2 à l'aide, par exemple, de la commande suivante :

```
# /etc/init.d/apache2 force-reload
```

Testons à présent la configuration d'Apache avec le module PHP activé en créant un fichier « `/var/www/html/test.php` » dont le contenu est :

```
<?  
phpinfo();  

```

Si nous tentons alors d'accéder à la page `http://194.10.20.2:8080/test.php` depuis le PC LAN, nous obtenons le résultat indiqué sur la Figure 3. Celui-ci nous indique que PHP4 est installé correctement et fonctionnel. Il décrit également la liste des modules PHP installés et activés, dont le module MySQL.

En effet, la commande `phpinfo()` permet d'afficher la configuration actuelle de l'interpréteur PHP et de vérifier la liste des modules chargés, les chemins utilisés, et les limitations fixées comme la taille maximale acceptée en upload.

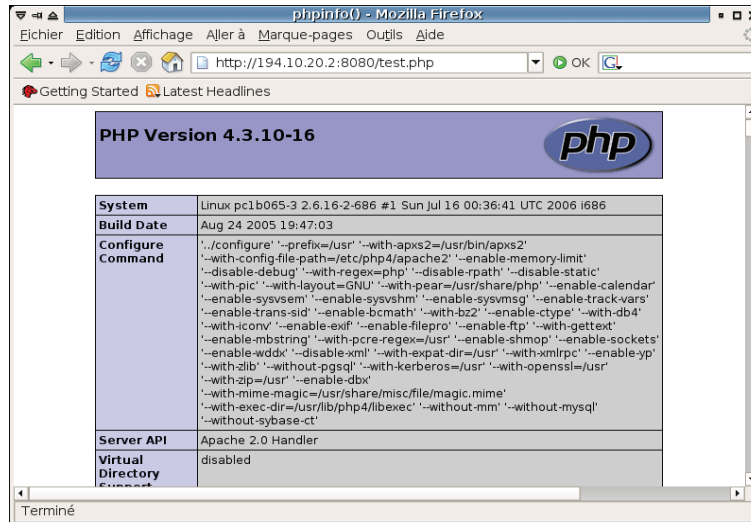


Figure 3 – Test de configuration du serveur Apache 2 avec le module PHP4 activé

3.2 Configuration d'un serveur DHCP

3.2.1 Introduction au protocole DHCP

DHCP signifie Dynamic Host Configuration Protocol et désigne un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une machine. Cela peut inclure son adresse IP, son masque de sous-réseau, son adresse de broadcast, l'adresse du réseau, ou encore l'adresse des routeurs et de la passerelle par défaut.

La station cliente qui souhaite se voir attribuer une adresse IP transmet un paquet UDP sur le réseau, en broadcast (255.255.255.255), adressé sur le port 67. Le(s) serveur(s) DHCP présents sur le réseau, qui écoutent sur leur port UDP 67, reçoivent ces paquets DHCP DISCOVER qui contiennent comme information minimale l'adresse Ethernet MAC du client et, éventuellement, une liste des paramètres demandés par le client.

Si un serveur accepte la requête, celui-ci propose une offre (message DHCP OFFER), envoyée en unicast Ethernet au client. Si le serveur accepte l'offre, il envoie, toujours en broadcast, un paquet DHCP REQUEST au serveur. Ce message permet également aux autres serveurs d'être informés que l'offre retenue n'est pas la leur. Si la transaction est confirmée par le serveur, un paquet DHCP ACK est envoyé en unicast Ethernet au client. Le client peut alors se configurer automatiquement à partir des paramètres reçus.

Comme les paquets transmis par le client sont envoyés en broadcast, ils ne sont pas propagés au-delà des limites du réseau local. Ainsi, un routeur ne doit pas relayer les paquets DHCP sur les autres réseaux avec lesquels il assure l'interconnexion.

3.2.2 Installation du serveur DHCP sur la machine DMZ

Comme nous l'avons vu précédemment, un serveur DHCP fournit un ensemble d'informations (adresse IP, masque, passerelle par défaut...) aux clients afin de leur permettre de se configurer. Dans la distribution Debian de Linux, plusieurs serveurs DHCP sont disponibles. Dans le cadre de ce TP, nous utilisons le serveur fourni dans le paquet Debian « `dhcp3-server` ».

```
# apt-get install dhcp3-server
```

La configuration du serveur DHCP se fait dans le fichier « `/etc/dhcp3/dhcpd.conf` ». Les principales de ce fichier de configuration sont :

- La directive « `option` » permet de spécifier les informations complémentaires à transmettre aux clients pour leur permettre de se configurer. On peut par exemple lui fournir :
 - le nom du domaine local avec « `option domain-name` »
 - les adresses serveurs de résolution de noms de domaine (serveurs DNS) avec « `option domain-name-servers` »
 - les adresses de passerelle par défaut avec « `option routers` »
 - l'adresse de broadcast du réseau avec « `option broadcast-address` »
- Les directives « `default-lease-time` » et « `max-lease-time` » permettent de définir la durée de vie de la configuration qui a été fournie au client. Cette durée dépend de l'infrastructure réseau qu'on veut mettre en place. Par exemple, dans un environnement de bureau, où les systèmes sont ajoutés et retirés peu fréquemment, une durée d'un mois ou plus peut avoir un sens. Généralement, une durée de quelques heures est utilisée (par exemple 24 heures)
- Les directives « `subnet` » et « `host` » sont en fait des blocs qui permettent de définir une configuration ciblée à un sous réseau (`subnet`) ou à une station (`host`).
 - Dans un bloc « `subnet` », on peut préciser la directive « `range` » qui donne l'intervalle d'adresses IP dans lequel on peut puiser pour configurer les clients. « `pool` » est une autre version utilisée en dehors de la déclaration `subnet`, plus facile à mettre en place de « `range` », et qui peut utiliser un filtrage des clients en conjonction avec une directive « `class` » et éventuellement des directives « `subclass` ».
 - Le bloc « `host` », permet d'assigner une configuration fixe à une machine du réseau. Il faut donc préciser son adresse MAC (« `hardware xx:xx:xx:xx:xx:xx` ») et l'adresse IP fixe à assigner (« `fixed-address` »).

Les directives précédentes sont des directives de paramétrage qui concernent tous les clients DHCP qui font une demande de configuration. Elles peuvent donc être précisées dans les blocs pour affiner leur configuration.

- La directive « `shared-network` » permet de regrouper plusieurs zones `subnet` ainsi que plusieurs `pool` d'adresse lorsque ceux-ci concernent le même réseau physique. Les paramètres spécifiés s'appliqueront donc aux sous réseaux englobés. Cette directive permet d'informer le serveur DHCP de la topologie du réseau et ainsi, de prendre en charge la présence de relais DHCP au sein des réseaux décrits.

Voici ci-dessous un extrait du fichier configuration qui a été utilisé pour mettre en place le serveur DHCP sur notre machine DMZ qui va se charger d'adresser le réseau DMZ et LAN tout en assurant des adresses fixes à certaines machines, des deux sous réseaux.

```
# Le bail DHCP est à 24 heures par défaut
default-lease-time 1440;
# Sa durée maximale est de 5 jours (120 heures)
max-lease-time 7200;

# définition d'une classe filtrant les clients par leur adresse MAC
class "mycompany-lan-computers" {
    match pick-first-value (option dhcp-client-identif, hardware);
}

# definition de sous classe, une par client autorisé
# elles seront utilisé dans un pool pour l'attribution dynamique
subclass "mycompany-lan-computers" 00:01:03:51:3A:4F;
subclass "mycompany-lan-computers" 00:01:03:51:3A:50;
# répéter une ligne par client autorisé

shared-network mycompany-com {

    # Definition du réseau DMZ
    subnet 194.10.20.0 netmask 255.255.255.0 {
        option routers 194.10.20.1;
        option broadcast-address 194.10.20.255;
        option domain-name "dmz.mycompany.com";
    }

    # Definition du réseau LAN
    subnet 194.10.10.0 netmask 255.255.255.0 {
        option routers 194.10.10.1;
        option broadcast-address 194.10.10.255;
        option domain-name "lan.mycompany.com";
    }
}
```



```
# option domain-name-servers 194.10.20.x <- S'il y avait un ou
# plusieurs DNS sur le réseau, ce qui n'est pas le cas

# permet de désactiver ip_forward sur les clients dhcp
option ip-forwarding false;
}

pool {
    # Non sécurisé mais plus simple à mettre en place
    allow unknown-clients;
    #Une façon plus sécurisée, grâce à une liste d'@ MAC ci-dessus
    #allow members of "mycompany-lan-computers";
    range 194.10.10.2 194.10.10.200;
}

# On peut aussi, par exemple, fixer des adresses en fonction des
# adresses MAC
host pc-lan-02 {
    hardware ethernet 00:01:03:55:66:77;
    fixed-address 194.10.10.2;
}

host pc-lan-03 {
    hardware ethernet 00:01:03:66:77:88;
    fixed-address 194.10.10.3;
}

host router-cote-dmz {
    hardware ethernet 00:01:03:42:CD:F2;
    fixed-address 194.10.20.1;
}

host serveur-dmz-2 {
    hardware ethernet 00:01:03:42:CF:F8;
    fixed-address 194.10.20.3;
}
}
```

Lançons notre serveur DHCP afin de voir si la configuration ci-dessus fonctionne avec la commande ci-dessous :

```
# /etc/init.d/dhcp3-server start
```

Ouvrons le fichier de log en parallèle afin de détecter les éventuelles erreurs.

```
# tail -f /var/log/syslog
```

Vérifions que notre serveur écoute bien sur le bon port avec la commande `netstat` en lui demandant d'afficher la liste des ports UDP ouverts (option `-u`). On filtre ensuite cette liste sur les lignes contenant la chaîne « 67 » avec la commande `grep`.

```
# netstat -uan | grep 67
udp        0      0 0.0.0.0:67          0.0.0.0:*
```

Notre serveur DHCP est donc configuré et prêt à être utilisé par les différents clients du réseau LAN et/ou DMZ (en cas de configuration de serveurs de la DMZ par DHCP, comme prévu dans le fichier de configuration ci-dessus).

Notons que dans la configuration décrite, n'importe quel client peut se voir attribuer une adresse IP dynamique par le serveur DHCP sur le réseau LAN. Pour sécuriser le pool d'adresses dynamiques pour des adresses MAC précises, il faut commenter la directive `allow unknown-clients;` et supprimer le commentaire de la ligne `allow members of "mycompany-lan-computers";`.

L'autorisation de nouveaux clients sur le serveur DHCP se ferait alors par l'ajout d'une ligne :

```
subclass "mycompany-lan-computers" <adresse MAC>;
```

A la suite de celles décrites dans le fichier de configuration.

3.2.3 Installation d'un relai DHCP sur la machine routeur

Comme cela est abordé dans la section 3.2, les demandes DHCP (`DHCP DISCOVER` et `DHCP REQUEST`) sont envoyées en broadcast. Par défaut, le routeur ne relaye donc pas les demandes DHCP hors du réseau sur lequel elles ont été émises.

Il faut donc également configurer le routeur pour qu'il soit relai des trames DHCP reçues en broadcast depuis le LAN, mais pas depuis le réseau DMZ ni le réseau internet.

Pour cela, il faut installer l'application `dhcp3-relay`, qui permet de rediriger les paquets DHCP reçus sur un réseau vers un serveur DMZ situé sur un autre réseau. Pour cela, nous utilisons la commande suivante, sous Debian :

```
# apt-get install dhcp3-relay
```

Pour configurer le relai, il faut modifier le script `/etc/default/dhcp3-relay` et lui mettre les valeurs correspondant à l'adresse IP du serveur DHCP (variable `SERVERS`) et les noms des interfaces sur lesquelles le relai doit écouter les échanges DHCP (variable `INTERFACES`) :

```
# What servers should the DHCP relay forward requests to?
SERVERS="194.10.20.2"

# On what interfaces should the DHCP relay (dhrelay) serve DHCP requests?
INTERFACES="eth0 eth1"
```

Le relai ne doit pas rediriger les paquets sur le réseau par lequel il reçoit les paquets, faute de quoi les paquets seront reçus en double par le serveur DHCP. En revanche, chaque interface réseau impliquée dans le service et le relai de requêtes DHCP doit être listée dans la variable `INTERFACES`, faute de quoi les paquets ne seront pas retransmis. C'est pour cela que les deux interfaces `eth0` et `eth1` ont été déclarées dans la variable `INTERFACES`.

Pour démarrer le service `dhcp relay v3`, il faut ensuite exécuter le script de démarrage suivant :

```
# /etc/init.d/dhcp3-relay start
```

Il convient de noter que le relai DHCP utilise le port UDP 67 sur l'interface `eth0` (celle où il écoute les demandes DHCP) et le port UDP 67 sur les interfaces qui le relient aux serveurs DHCP vers lesquels il redirige les requêtes. Ces informations sont importantes pour la configuration de la sécurité dans la section 4.3.2.

3.2.4 Configuration du client DHCP sur le PC LAN

Pour tester notre serveur DHCP, nous allons démarrer le client DHCP à partir du PC LAN. Nous utiliserons ensuite l'outil `ethereal` sur notre relai DHCP (la machine routeur) afin de capturer puis d'analyser les échanges de données occasionnés.

Pour que le client DHCP démarre automatiquement à chaque fois que l'interface est activée il faut modifier le fichier `/etc/network/interfaces`. On indique donc dans ce fichier que cette interface doit être configurée dynamiquement via DHCP.

```
auto eth0
iface eth0 inet dhcp
```

La première ligne permet de préciser les noms des interfaces qui doivent être démarrées et configurées automatiquement au démarrage du système, ici `eth0`. La seconde ligne permet de configurer l'interface (`iface`) `eth0` pour utiliser un réseau IPv4 (`inet`) et pour qu'elle reçoive sa configuration d'un serveur `dhcp` (à l'aide de `dhclient` ou `dhclient3`).

Pour désactiver l'interface et la réactiver en lançant le client DHCP automatiquement, il suffira alors d'exécuter les commandes :

```
# ifconfig eth0 down  
# ifup eth0
```

Il est cependant possible de lancer manuellement et à n'importe quel moment le client DHCP en tapant la commande suivante :

```
# dhclient eth0
```

ou

```
# dhclient3 eth0
```

Attention cependant à ne pas avoir plusieurs instances de `dhclient` fonctionnant pour configurer la même carte réseau. Vérifiez qu'au plus un seul processus est lancé par carte à l'aide de la commande `ps`. Au besoin, utilisez `kill -9 process_id` pour supprimer les clients DHCP superflus à l'aide des `PID` fournis par la commande `ps`.

Le client DHCP peut également être configuré plus précisément, pour demander et/ou exiger certains paramètres, et pour modifier les valeurs des `timeout` en cas de non réponse.

Cette configuration peut se faire dans le fichier `/etc/dhclient.conf` ou `/etc/dhcp3/dhclient.conf` suivant la version de `dhclient` utilisée.

Dans les tests de la section suivante, nous aurons besoin de modifier la durée du bail DHCP. Pour la modifier, il est conseillé de modifier l'option `default-lease-time` du serveur DHCP et de la mettre à `1` au lieu de `1440`. Cela permet d'étudier le processus de renouvellement du bail DHCP. Pour que l'information soit prise en compte par le client, il suffit de couper l'interface réseau (`ifdown eth0`) et de la réactiver (`ifup eth0`).

3.2.5 Tests de la configuration : Assignment et renouvellement

✘ Assignment des paramètres TCP/IP

Voici les trames capturées avec `ethereal` sur le PC routeur après démarrage du client sur le PC LAN :

| No. - | Time | Source | Destination | Protocol | Info |
|-------|-----------|-------------------|-----------------|----------|---|
| 1 | 0.000000 | Intel-HF_ce:04:19 | | ARP | who has 172.17.0.72? Tell 172.17.0.254 |
| 2 | 0.006837 | HP_51:97:00 | | ARP | 172.17.0.72 is at 08:00:09:51:97:00 |
| 3 | 16.188601 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID 0x72753357 |
| 4 | 16.192227 | 3com_42:cd:f2 | | ARP | who has 194.10.20.2? Tell 194.10.20.1 |
| 5 | 16.192362 | DellComp_96:cf:98 | | ARP | 194.10.20.2 is at 00:06:5b:96:cf:98 |
| 6 | 16.192379 | 194.10.20.1 | 194.10.20.2 | DHCP | DHCP Discover - Transaction ID 0x72753357 |
| 7 | 16.192768 | 194.10.20.2 | 194.10.10.200 | ICMP | Echo (ping) request |
| 8 | 16.196221 | 3com_09:4d:c2 | | ARP | who has 194.10.10.200? Tell 194.10.10.1 |
| 9 | 16.371642 | 194.10.20.2 | 194.10.10.1 | DHCP | DHCP Offer - Transaction ID 0x72753357 |
| 11 | 16.372470 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID 0x72753357 |
| 12 | 16.372608 | 194.10.20.1 | 194.10.20.2 | DHCP | DHCP Request - Transaction ID 0x72753357 |
| 13 | 16.375196 | 194.10.20.2 | 194.10.10.1 | DHCP | DHCP ACK - Transaction ID 0x72753357 |
| 15 | 17.196295 | 3com_09:4d:c2 | | ARP | who has 194.10.10.200? Tell 194.10.10.1 |
| 16 | 17.196438 | DellComp_c4:44:24 | | ARP | 194.10.10.200 is at 00:06:5b:c4:44:24 |
| 17 | 17.196462 | 194.10.20.2 | 194.10.10.200 | ICMP | Echo (ping) request |
| 18 | 17.196562 | 194.10.10.200 | 194.10.20.2 | ICMP | Echo (ping) reply |
| 19 | 17.196627 | 194.10.10.200 | 194.10.20.2 | ICMP | Echo (ping) reply |

Figure 4 - Echanges DHCP observés au niveau du relai

Lorsque le client DHCP démarre, il n'a en principe aucune connaissance du réseau.

Il envoie donc une trame "DHCPDISCOVER", dont le but est de localiser les serveurs DHCP disponibles et de demander une première configuration. Cette trame est un "broadcast", donc envoyé à l'adresse 255.255.255.255.

N'ayant pas encore d'adresse IP, il adopte provisoirement l'adresse IP 0.0.0.0. Comme ce n'est pas avec cette adresse que le DHCP va l'identifier, il fournit aussi sa "MAC Address" Ethernet.

Le relai DHCP retransmet cette trame au serveur DHCP, en unicast, et en lui ajoutant une information indiquant la présence du relai. Lorsque le serveur DHCP du réseau reçoit cette trame, il décide d'offrir l'adresse 194.10.10.200. Il commence alors par effectuer un ping (trame ICMP) de cette adresse afin de vérifier la disponibilité de celle-ci.

Etant donné qu'aucune réponse n'a été apportée à ce ping, il en déduit que cette adresse est disponible et répond alors par un "DHCPOFFER" pour envoyer son offre au relai DHCP, en unicast IP. Le paquet est ensuite retransmis au demandeur, en unicast Ethernet à l'aide de l'adresse MAC reçue du client.

Cette trame contient une proposition de bail (durée de validité ainsi que d'autres informations de configuration) et la "MAC Address" du client, avec également l'adresse IP du serveur.

Dans le cas où plusieurs serveurs DHCP se trouveraient dans un même réseau, tous répondent et le client accepte la réponse qui correspond le mieux aux paramètres attendus, précisés dans le fichier `/etc/dhclient.conf`.

Le client répond alors par un `DHCPREQUEST` envoyé en broadcast (il n'a toujours pas d'adresse IP) sur le réseau de manière à indiquer quelle offre a été acceptée. Le relai retransmet ce message au serveur, toujours en lui ajoutant une information sur la présence du relai DHCP.

Le serveur DHCP Concerné répond définitivement par un `DHCPACK`, envoyé en unicast au relai DHCP, qui constitue une confirmation du bail. Le relai retransmet cette information en unicast Ethernet au demandeur. Celui-ci se configure alors en fonction des paramètres reçus.

L'adresse du client est alors marquée comme utilisée par le serveur DHCP et ne sera plus proposée à un autre client pour toute la durée du bail, à moins que le client ne mette fin au bail en envoyant un paquet `DHCPRELEASE`.

✘ **Renouvellement de bail DHCP**

Lorsque la durée du bail est inférieure à " l'uptime" du client, autrement dit, si le client (en l'occurrence le PC LAN) reste connecté plus longtemps que la durée de validité de son bail, il va devoir le renouveler.

Pour visualiser la procédure de renouvellement, nous diminuons notre timeout à 1 minute dans le fichier de configuration du serveur, comme indiqué précédemment.

Nous constatons en analysant les échanges de données qu'au bout de 30 secondes (soit 50% de la durée de vie du bail), le client envoi une requête de type `DHCPREQUEST` au serveur, en précisant sa configuration actuelle.

Ce dernier a aussitôt répondu par un `DHCPACK`.

Dans le cas où le serveur DHCP n'aurait pas répondu à cette requête, le client tentera une nouvelle fois de renouveler son bail à 75% puis 87,5% de la durée de validité du bail.

Si à 87,5%, le client n'a toujours pas de réponse il considère que le serveur est en panne et envoi alors un `DHCPDISCOVER` en broadcast pour espérer obtenir une réponse auprès d'un autre serveur DHCP.

S'il reçoit un `DHCPOFFER`, alors le client mettra son bail à jour et effectuera son prochain renouvellement auprès de ce nouveau serveur.

3.3 Test de la sécurité pour le PC de la DMZ

Afin de tester les ports accessibles depuis le PC du réseau DMZ (et, à fortiori, depuis n'importe quel PC de n'importe quel réseau vers ce PC), on peut utiliser l'utilitaire `nmap`. Celui-ci n'est pas installé par une installation classique Debian. Pour l'installer, il suffit alors d'utiliser `aptitude` en exécutant la commande suivante :

```
# apt-get install nmap
```

On peut ensuite utiliser `nmap` pour scanner les ports TCP ouverts sur la machine située sur le DMZ, qui a pour adresse IP `194.10.20.2`, comme suit :

```
# nmap 194.10.20.2

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2006-10-04 16:08
CEST
Interesting ports on 194.10.20.2:
(The 1658 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
113/tcp   open  auth
927/tcp   open  unknown
8080/tcp  open  http-proxy
MAC Address: 00:06:5B:96:CF:98 (Dell Computer)

Nmap finished: 1 IP address (1 host up) scanned in 10.538 seconds
```

On fait de même pour lister les ports UDP détectés par `nmap` comme étant ouverts, comme suit :

```
nmap -sU -F 10.2.28.31

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2006-10-04 16:12
CEST
Interesting ports on 194.10.20.2:
(The 1004 ports scanned but not shown below are in state: closed)
PORT      STATE      SERVICE
67/udp    open|filtered dhcpserver
```

On remarque que tous les ports ouverts sont facilement détectables: le port UDP du serveur DHCP (`67/udp`), le port d'administration distante SSH (`22/tcp`), le port `identd` (`113/tcp`), le port non conventionnel utilisé pour notre serveur http (`8080/tcp`), ainsi qu'un port dont le rôle n'est pas connu (`927/tcp`).

Afin de protéger au mieux les réseaux LAN et DMZ tout en leur laissant un accès à internet et en permettant au DMZ de proposer ses services aux ordinateurs des réseaux extérieurs, il est primordial de configurer un pare-feu (firewall) sur le routeur.

4 Sécurisation de la plateforme

4.1 Introduction au firewall statefull iptables

« `iptables` » est un logiciel libre de firewalling utilisé pour configurer les tables des règles de filtrage des paquets IP dans le noyau linux. Ce module ainsi que les modules correspondant à ces tables devront être activés s'ils n'ont pas été compilés dans le noyau. L'activation se fait avec les commandes ci-dessous :

```
#modeprobe ip_tables
#modeprobe iptable_filter
#modeprobe iptable_nat
#modeprobe iptable_mangle
#modeprobe ipt_MASQUERADE
```

Par défaut sous Debian Sarge, ces modules sont compilés dans le noyau.

Plusieurs tables peuvent être définies. Chaque table contient un certain nombre de chaînes prédéfinies. Chaque chaîne est une liste de règles qui peuvent concorder avec les paquets IP. Chaque règle spécifie ce qui doit être fait avec le paquet qui concorde. Cela est appelé une cible, laquelle correspond à un saut vers une chaîne utilisateur dans la même table.

A travers le traitement de ces paquets, « `iptables` » facilite la mise en place des deux types de politique d'accès au réseau qui nécessitent bien entendu la participation active des utilisateurs :

- Tout permettre, sauf ce qui est explicitement interdit
- Tout interdire sauf ce qui est explicitement permis

Qu'une architecture de réseau soit simple (simple filtrage des connexions entrantes et sortantes) ou complexe (mise en place de proxy ou de DMZ), un firewall doit être capable de réaliser les actions suivantes :

- Utilisation de règles de filtrage qu'on définit en concordance avec la politique d'accès au réseau.
- translations de ports et d'adresses qui peuvent être dynamiques ou statiques.

Dans « `iptables` », ces deux actions se traduisent par deux tables (respectivement `NAT` et `FILTER`). Ces tables admettent différentes chaînes. Une chaîne est une suite de règles prises dans l'ordre. Dès qu'une règle s'applique à un paquet, elle est déclenchée, et la suite de la chaîne est ignorée. Ces règles sont souvent complétées par des cibles qui permettent de préciser ce que fait une règle donnée.

La table `NAT` (Network Address Translation) est utilisée pour la translation d'adresse ou la translation de port. Elle a deux types de chaînes : `PREROUTING` qui permet de spécifier

« à l'arrivée du firewall » et la chaîne `POSTROUTING` qui permet de spécifier « à la sortie du firewall ». Il existe trois cibles : `DNAT`, `SNAT` et `MASQUERADE`.

La table `FILTER` est la table par défaut lorsque l'on n'en spécifie pas. Cette table contient toutes les règles de filtrage, il existe trois types de chaînes : `FORWARD` pour les paquets passant par le firewall, `INPUT` pour les paquets entrant et `OUTPUT` pour les paquets sortants. Les cibles disponibles sont : `ACCEPT`, `DENY`, `DROP`, `REJECT`.

4.2 Sécurisation des communications par restriction

Afin de garantir une plus grande sécurité, nous avons décidé de mettre en place une stratégie de protection restrictive : toutes les communications entre réseaux sont interdites à moins qu'elles n'aient été explicitement autorisées.

Pour cela, il suffit de changer la politique de traitement par défaut des paquets passant par les chaînes `INPUT`, `FORWARD` ou `OUTPUT` en mode `DROP`, `DENY` ou `REJECT`.

Dans notre cas, nous avons décidé d'utiliser la politique `DROP` par défaut, car celle-ci n'analyse pas les paquets au-delà de la couche 2 du modèle OSI et car elle n'informe par l'émetteur du paquet de la suppression de celui-ci.

Ce choix présente trois avantages :

- Rendre le traitement des paquets rejetés plus rapide,
- Masquer partiellement aux autres clients l'existence de services/machines,
- Ne pas encombrer les réseaux avec des paquets de retour d'erreur.

Ce dernier point est particulièrement intéressant si une des connexions (par exemple la connexion internet) possède une faible bande passante, facilement saturable.

Pour mettre en place une telle stratégie, les commandes suivantes doivent être utilisées :

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Remarque : la commande `iptables -F` ne suffira alors plus à réinitialiser l'état des chaînes de traitement des paquets IP. Il faudra utiliser `iptables -P CHAIN_NAME ACCEPT` pour rétablir la politique des chaînes à une acceptation par défaut.

Si on consulte alors la table de routage par défaut et la table de routage NAT à l'aide des commandes `iptables -L` et `iptables -t nat -L`, on obtient les informations de la page suivante.

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination

Chain FORWARD (policy DROP)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination

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

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination

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

On observe que la politique par défaut des chaînes `INPUT`, `OUTPUT` et `FORWARD` de la table de routage `filter` (table par défaut) est bien `DROP`. En revanche, la politique par défaut des chaînes `PREROUTING`, `POSTROUTING` et `OUTPUT` de la table de routage `nat` est `ACCEPT`.

Ce comportement n'est pas gênant dans la mesure où ces règles s'appliquent avant et/ou après la règle `FORWARD` de la table `filter`. De plus, aucune translation d'adresse ni de port n'a été configurée pour le moment car les chaînes ne comportent aucune entrée. Il faudra cependant être vigilant lors de la configuration NAT/PAT des sections 4.4 et 4.5.

On peut s'assurer que cette configuration fonctionne en tentant d'envoyer des requêtes ICMP ping entre la DMZ et le LAN et réciproquement. Ainsi, depuis le PC de la DMZ :

```
# ping 194.10.10.2
PING 194.10.10.2 (194.10.10.2) 56(84) bytes of data.

--- 194.10.10.2 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2011ms
```

Et depuis le PC connecté au LAN :

```
# ping 194.10.20.2
PING 194.10.20.2 (194.10.20.2) 56(84) bytes of data.

--- 194.10.20.2 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3010ms
```

Aucune communication n'est alors possible entre les zones DMZ, LAN et Internet. Le réseau est donc sécurisé par restriction.

4.3 Filtrage interne

La sécurisation des échanges entre les différents réseaux (LAN, DMZ et Internet) doit se faire de façon transparente. Les services utiles proposés par la DMZ à destination du réseau local ne doivent pas être restreint par l'utilisation d'un pare-feu.

Il nous faut donc autoriser les trafics ICMP (requis pour le ping), http (sur le port TCP non conventionnel 8080) et DHCP (sur les ports UDP conventionnel 67 et 68) entre la zone démilitarisée et le réseau local.

4.3.1 Support des paquets ICMP entre LAN, routeur et DMZ

Dans un premier temps, on configure le routeur pour qu'il réponde aux requêtes ping (protocole ICMP) sur l'interface connectée au réseau local :

```
iptables -A INPUT -p icmp -s 194.10.10.0/24 -d 194.10.10.1 -j ACCEPT
iptables -A OUTPUT -p icmp -s 194.10.10.1 -d 194.10.10.0/24 -j ACCEPT
```

Les ordinateurs du réseau LAN peuvent désormais envoyer des paquets ICMP à l'interface LAN du routeur, et le routeur peut leur répondre. Par exemple, depuis la machine 194.10.10.2 du LAN :

```
PING 194.10.10.1 (194.10.10.1) 56(84) bytes of data.
64 bytes from 194.10.10.1: icmp_seq=1 ttl=64 time=0.222 ms
64 bytes from 194.10.10.1: icmp_seq=2 ttl=64 time=0.183 ms

--- 194.10.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.183/0.202/0.222/0.024 ms
```

On fait la même chose pour l'interface du routeur connectée au réseau DMZ :

```
iptables -A INPUT -p icmp -s 194.10.20.0/24 -d 194.10.20.1 -j ACCEPT
iptables -A OUTPUT -p icmp -s 194.10.20.1 -d 194.10.20.0/24 -j ACCEPT
```

Bien que les requêtes ICMP soient servies par les deux interfaces, cette configuration ne suffit pas à permettre à un ordinateur du réseau local d'envoyer des paquets ICMP à une machine de la zone démilitarisée. Pour que cela soit possible, il faut configurer la chaîne `FORWARD` pour qu'elle route correctement ce type de paquets. Cela se fait par l'ajout des règles suivantes :

```
iptables -A FORWARD -p icmp -s 194.10.10.0/24 -d 194.10.20.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -s 194.10.20.0/24 -d 194.10.10.0/24 -j ACCEPT
```

Avec `ethereal` ou, si la machine du réseau DMZ est dénuée d'interface graphique, avec `tcpdump`, on peut observer les paquets qui sont échangés entre un ordinateur et un autre.

Dans notre cas, le PC DMZ ne possédait pas d'interface graphique. Aussi, nous avons utilisé l'outil `tcpdump` pour capturer les paquets, puis nous avons visualisé le résultat sur une autre machine à l'aide d'`ethereal`. Le résultat obtenu suite au `ping` du PC de la DMZ par le PC du LAN est indiqué sur la Figure 5 ci-après.

| | | | | | |
|---|----------|-------------|-------------|------|---------------------|
| 1 | 0.000000 | 194.10.10.2 | 194.10.20.2 | ICMP | Echo (ping) request |
| 2 | 0.000014 | 194.10.20.2 | 194.10.10.2 | ICMP | Echo (ping) reply |
| 3 | 0.999993 | 194.10.10.2 | 194.10.20.2 | ICMP | Echo (ping) request |
| 4 | 1.000009 | 194.10.20.2 | 194.10.10.2 | ICMP | Echo (ping) reply |
| 5 | 2.000069 | 194.10.10.2 | 194.10.20.2 | ICMP | Echo (ping) request |

Figure 5 - Capture `ethereal` des échanges ICMP entre le PC LAN et le PC DMZ

Si on observe de plus près un de ces paquets, par exemple le paquet n°1 (réception d'une requête `ping`), On observe que les adresses IP concernées par l'échange sont les adresses réelles des machines, ce qui correspond à ce que nous souhaitons puisqu'il n'y a pas de traduction d'adresse (NAT) qui soit utilisée pour le moment.

Cependant, on peut noter que l'adresse MAC Source de ce paquet est l'adresse MAC de l'interface du routeur connectée au LAN, et non l'adresse MAC de l'interface `eth0` du PC de la zone démilitarisée.

En effet, lors d'un routage TCP/IP, les routeurs remplacent les adresses MAC source et/ou destination des paquets routés pour utiliser des adresses toujours valides et directement accessibles sur le réseau sur lequel ils sont situés. C'est le fonctionnement classique d'un matériel Ethernet interconnectant plusieurs machines.

4.3.2 Support du relai DHCP

Le principe de fonctionnement du relai DHCP est le suivant : il accepte les communications reçues en broadcast ou unicast depuis le réseau local vers le port UDP/67 (serveur DHCP) et retransmet les demandes DHCP reçues vers le serveur DHCP du DMZ via son port UDP/67, à destination du port UDP/67.

Le serveur DHCP est informé que le paquet provient d'un relai, car un en-tête supplémentaire est ajouté dans le paquet.

C'est cet en-tête qui permet au serveur DHCP de savoir quel est le sous réseau concerné par la demande et ainsi :

- d'attribuer une adresse IP dans le bon pool d'adresses,
- de donner la configuration IP correspondant au réseau concerné.

Pour que le relai DHCP fonctionne malgré la présence du pare-feu, il est nécessaire d'ajouter un certain nombre de règles `iptables` :

```
iptables -A FORWARD -p udp --sport 67 -s 194.10.20.2 -d 194.10.10.1 -j  
ACCEPT  
iptables -A INPUT -p udp --dport 67 --sport 67 -s 194.10.20.2 -d 194.10.10.1  
  
iptables -A INPUT -p udp --dport 67 --sport 68 -s 194.10.10.0/24 -d  
194.10.10.1  
iptables -A OUTPUT -p udp --sport 67 --dport 68 -s 194.10.10.1 -d  
194.10.10.0/24
```

La première règle autorise les paquets reçus sur le port UDP 67 de l'interface ayant pour adresse IP `194.10.10.1` à être pris en compte par le routeur et, par conséquent, par son relai DHCP. Elle est indispensable pour la réception des réponses du serveur DHCP comme les messages `DHCPOFFER` et `DHCPACK`.

La seconde règle autorise les paquets transmis du port UDP 67 du serveur DHCP vers le port UDP 67 de l'adresse `194.10.10.1` à être acceptés par le routeur. Ainsi, ils seront traités par `dhcp-relay3`.

La troisième règle autorise les clients DHCP du LAN (réseau `194.10.10.0/24`) à envoyer depuis leur port UDP 68 des paquets DHCP sur le port UDP 67 du serveur Routeur, pour que celui-ci soit pris en compte par le service `dhcp3-relay`.

Enfin, la dernière règle autorise le routeur à réémettre, sur le port UDP 67 de l'interface ayant pour adresse `194.10.10.1`, les réponses DHCP renvoyées par le serveur DHCP de la DMZ, tel qu'elles ont été retraitées par le relai DHCP.

Note : une règle `FORWARD` et `INPUT` sont indispensables pour chaque serveur DHCP vers lesquels les paquets doivent être relayés. Les autres règles sont correctes pour servir n'importe quels PC du LAN.

4.3.3 Support des autres services de la DMZ

La zone démilitarisée propose un serveur web http sur le port 8080 de la machine ayant pour adresse IP 194.10.20.2. Pour que les machines du réseau local puissent accéder à ce service en utilisant l'adresse IP et le numéro de port de la machine, il faut que le routeur autorise ce type de trafic. Pour cela, il faut les règles iptables suivantes dans la chaîne FORWARD :

```
iptables -A FORWARD -p tcp --dport 8080 -s 194.10.10.0/24 -d 194.10.20.2 -j ACCEPT
iptables -A FORWARD -p tcp --sport 8080 -s 194.10.20.2 -d 194.10.10.0/24 -j ACCEPT ! --syn
```

Ces règles autorisent respectivement le trafic depuis le réseau LAN vers le port 8080 du PC de la DMZ, et le trafic du port 8080 du PC de la DMZ à destination du réseau LAN, du moment que ce ne sont pas des demandes de connexion (bit SYN du protocole TCP).

On teste alors depuis un PC du réseau local, et on obtient bien la page d'accueil du site hébergé sur la DMZ à partir de l'url <http://194.10.20.2:8080/>, comme présenté sur la figure ci-après.

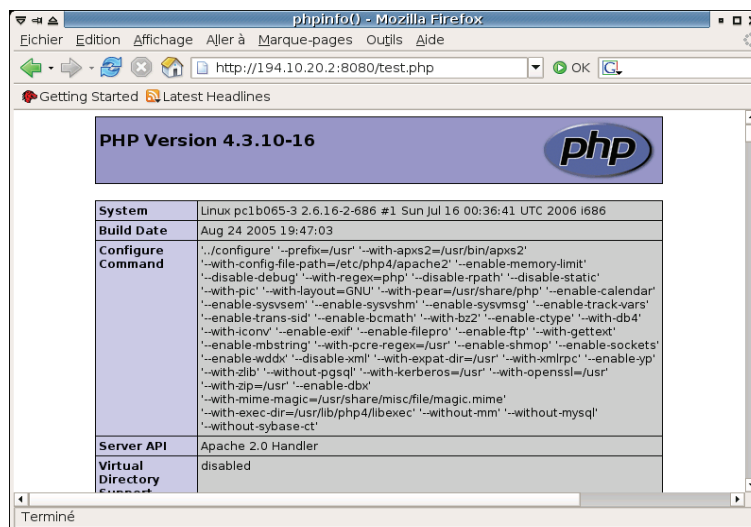


Figure 6 - Résultat d'une connexion à l'url <http://194.10.20.2:8080/test.php> depuis le LAN avec Firefox

L'accès direct au serveur web depuis le LAN est donc opérationnel. Il pourrait également être intéressant, à des fins d'administration, d'autoriser les connexions au port SSH (22/tcp) du PC de la DMZ depuis le réseau local. Ceci permettrait en effet aux administrateurs de la machine d'intervenir à distance depuis le réseau LAN en cas de problèmes. Ceci peut, en revanche, être dangereux dans le cas où une personne mal intentionnée réussirait à se connecter à une machine du LAN.

4.4 Translation d'adresses (NAT)

Afin que les machines du réseau local ne soient pas accessibles directement depuis internet, il est préférable d'utiliser une traduction d'adresses. C'est un mécanisme qui permet de faire correspondre les adresses IP internes d'une organisation vers un ensemble d'adresses externes, qui seront les seules visibles au public.

Ce procédé permet, entre autre, d'utiliser un adressage privé au sein du réseau de l'entreprise et de n'avoir qu'un nombre limité d'adresses IP publiques accessibles depuis internet. Dans la mesure où l'adressage IPv4 ne permet l'utilisation que de 2^{32} adresses au maximum, il est souvent rare que les entreprises aient suffisamment d'adresses IP publiques pour connecter l'ensemble de leurs matériels sur des IP publiques. Ce procédé est donc très utilisé.

Pour activer le NAT depuis le réseau local (194.10.10.0/24) vers les réseaux extérieurs, via l'interface `eth2`, il est donc nécessaire d'ajouter la règle `iptables` suivante :

```
iptables -A POSTROUTING -t nat -s 194.10.10.0/24 -o eth2 -j MASQUERADE
```

Cette règle ne suffira cependant pas à donner accès aux services extérieurs depuis le réseau local, car aucune règle de transmission de paquet (`FORWARD`) n'autorise le trafic vers l'extérieur.

Il convient alors de rédiger un inventaire des services qui doivent être accessibles depuis le réseau local à destination de l'internet, afin d'appliquer une stratégie la plus sécuritaire possible. De plus, le trafic ne doit être autorisé que dans un seul sens, afin d'éviter les intrusions extérieures sur le réseau local.

Les services que nous souhaitons autoriser sont les suivants :

- Accès aux sites web extérieurs – sur le port 80
- Accès aux serveurs SSH extérieurs – sur le port 22
- Possibilité d'émettre des requêtes ping et d'en recevoir les réponses.

Les règles `iptables` suivantes permettent d'autoriser, respectivement, ces trafics :

```
iptables -A FORWARD -p tcp --dport 80 -s 194.10.10.0/24 -o eth2 -j ACCEPT
iptables -A FORWARD -p tcp --sport 80 -i eth2 -d 194.10.10.0/24 -j ACCEPT !
--syn

iptables -A FORWARD -p tcp --dport 22 -s 194.10.10.0/24 -o eth2 -j ACCEPT
iptables -A FORWARD -p tcp --sport 22 -d 194.10.10.0/24 -i eth2 -j ACCEPT !
--syn

iptables -A FORWARD -p icmp -s 194.10.10.0/24 -o eth2 -j ACCEPT
iptables -A FORWARD -p icmp -i eth2 -d 194.10.10.0/24 --icmp-type echo-reply
-j ACCEPT
```

La première ligne autorise les ordinateurs du réseau local à accéder à n'importe quelle adresse IP internet (interface eth2) sur le port 80 – c'est-à-dire le port HTTP. La seconde autorise ce trafic dans l'autre sens, uniquement.

Les règles 3 et 4 autorisent un trafic similaire pour le protocole SSH (port tcp 22). Les deux dernière règles permettent d'autoriser l'envoi de paquets ICMP depuis le réseau LAN vers l'internet, et la suivante d'autoriser la réception de réponses `echo-reply` (réponse à un `ping`) depuis l'internet vers le réseau local.

Les requêtes `ping` seront refusées des réseaux extérieurs vers le routeur, ce qui n'est pas conformes aux RFC concernant le protocole IP. Pour être conforme, il faut que le routeur réponde aux requêtes `ping`, sur l'interface accessible depuis l'extérieur (`eth2`). Nous ajoutons donc les règles suivantes :

```
iptables -A INPUT -p icmp -s 0.0.0.0/0 -d 194.10.30.1 -j ACCEPT
iptables -A OUTPUT -p icmp -s 194.10.30.1 -d 0.0.0.0/0 -j ACCEPT
```

Ces règles autoriseront les réponses aux requêtes ping provenant du réseau LAN, de la DMZ et de l'internet vers l'interface ayant pour adresse `194.10.30.1`. Si l'interface a une adresse IP dynamique (par exemple une connexion PPP ou PPPoE), alors il faudra utiliser `-i eth2` au lieu de `-s 194.10.30.1` et `-o eth2` au lieu de `-d 194.10.30.1`.

Durant le TP nous n'avons pu tester l'ensemble de ces règles, pour des raisons de temps. Cependant de nombreuses documentations sur le firewall `iptables` nous ont permis de vérifier le fonctionnement théorique de l'ensemble de ces règles.

4.5 Translation de ports (PAT)

4.5.1 Mise en place d'une translation de port

Afin que les services web proposés par la zone démilitarisée ne soient accessibles que par l'adresse IP du routeur, sur le port 80 de celui-ci, il est nécessaire de faire une translation de port (PAT – Port Address Translation).

Dans un premier temps, il faut autoriser les échanges sur le port 80 du routeur depuis et à destination de toutes les machines (autres machines de la DMZ, LAN, et machines internet). Pour cela, on utilise les deux règles suivantes :

```
iptables -A INPUT -p tcp --sport 80 -s 0.0.0.0/0 -d 194.10.30.1 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -s 194.10.30.1 -d 0.0.0.0/0 -j ACCEPT
```

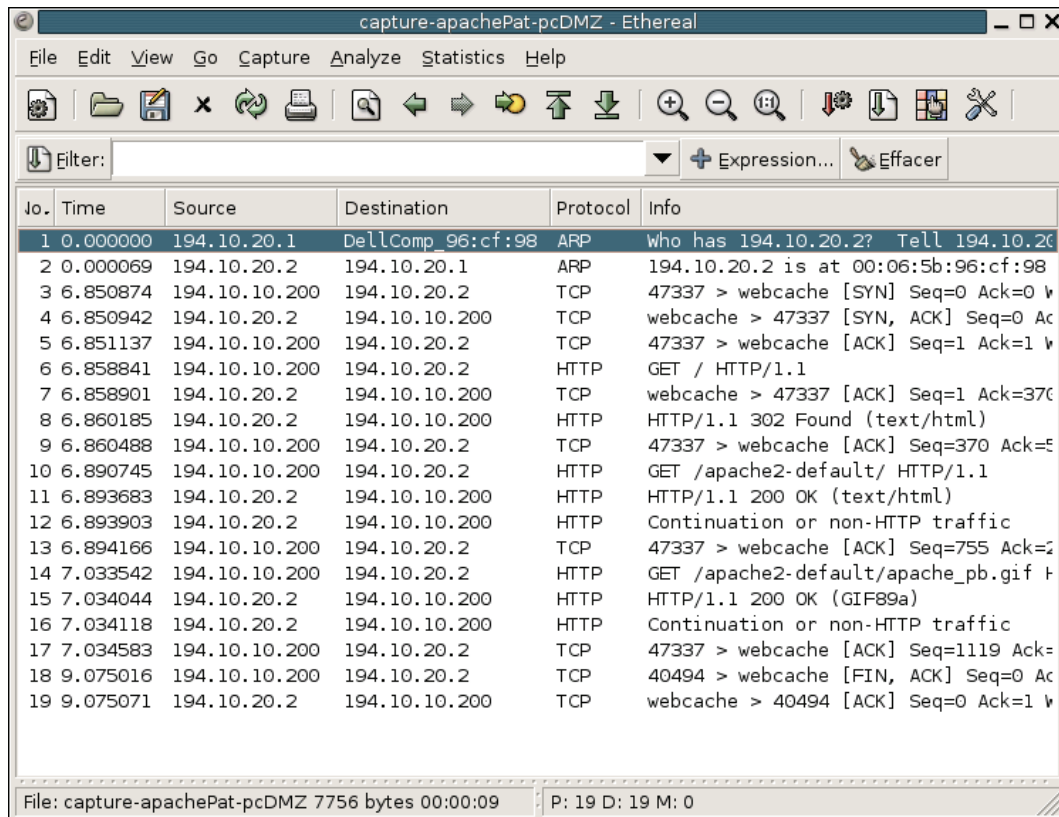

Puis, il faut réaliser la traduction de port (PAT) entre le port 80 routeur et le port 8080 de la machine 194.10.20.2 située dans la zone démilitarisée. Pour cela, il faut appliquer les règles suivantes à la table de routage NAT :

```
iptables -A PREROUTING -t nat -d 194.10.30.1 -p tcp --dport 80 -j DNAT --to-destination 194.10.20.2:8080
iptables -A FORWARD -i eth2 -d 194.10.20.2 -dport 8080 -j ACCEPT
```

4.5.2 Tests de fonctionnement de la translation de ports

Afin de tester la translation de port, nous tentons de visualiser les paquets échangés entre un client quelconque (par exemple un PC du LAN ou un PC de la zone internet) et le PC de la DMZ qui fournit le service web, via l'URL <http://194.10.30.1/>.

Voici la capture `ethereal` réalisé sur le PC DMZ, qui permet d'observer les paquets échangés entre le client et le serveur web, via la translation de port :



| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|---------------|-------------------|----------|---|
| 1 | 0.000000 | 194.10.20.1 | DellComp_96:cf:98 | ARP | Who has 194.10.20.2? Tell 194.10.20.2 |
| 2 | 0.000069 | 194.10.20.2 | 194.10.20.1 | ARP | 194.10.20.2 is at 00:06:5b:96:cf:98 |
| 3 | 6.850874 | 194.10.10.200 | 194.10.20.2 | TCP | 47337 > webcache [SYN] Seq=0 Ack=0 Win=0 Len=0 |
| 4 | 6.850942 | 194.10.20.2 | 194.10.10.200 | TCP | webcache > 47337 [SYN, ACK] Seq=0 Ack=47337 Win=0 Len=0 |
| 5 | 6.851137 | 194.10.10.200 | 194.10.20.2 | TCP | 47337 > webcache [ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 6 | 6.858841 | 194.10.10.200 | 194.10.20.2 | HTTP | GET / HTTP/1.1 |
| 7 | 6.858901 | 194.10.20.2 | 194.10.10.200 | TCP | webcache > 47337 [ACK] Seq=1 Ack=370 Win=0 Len=0 |
| 8 | 6.860185 | 194.10.20.2 | 194.10.10.200 | HTTP | HTTP/1.1 302 Found (text/html) |
| 9 | 6.860488 | 194.10.10.200 | 194.10.20.2 | TCP | 47337 > webcache [ACK] Seq=370 Ack=511 Win=0 Len=0 |
| 10 | 6.890745 | 194.10.10.200 | 194.10.20.2 | HTTP | GET /apache2-default/ HTTP/1.1 |
| 11 | 6.893683 | 194.10.20.2 | 194.10.10.200 | HTTP | HTTP/1.1 200 OK (text/html) |
| 12 | 6.893903 | 194.10.20.2 | 194.10.10.200 | HTTP | Continuation or non-HTTP traffic |
| 13 | 6.894166 | 194.10.10.200 | 194.10.20.2 | TCP | 47337 > webcache [ACK] Seq=755 Ack=211 Win=0 Len=0 |
| 14 | 7.033542 | 194.10.10.200 | 194.10.20.2 | HTTP | GET /apache2-default/apache_pb.gif HTTP/1.1 |
| 15 | 7.034044 | 194.10.20.2 | 194.10.10.200 | HTTP | HTTP/1.1 200 OK (GIF89a) |
| 16 | 7.034118 | 194.10.20.2 | 194.10.10.200 | HTTP | Continuation or non-HTTP traffic |
| 17 | 7.034583 | 194.10.10.200 | 194.10.20.2 | TCP | 47337 > webcache [ACK] Seq=1119 Ack=40494 Win=0 Len=0 |
| 18 | 9.075016 | 194.10.10.200 | 194.10.20.2 | TCP | 40494 > webcache [FIN, ACK] Seq=0 Ack=47337 Win=0 Len=0 |
| 19 | 9.075071 | 194.10.20.2 | 194.10.10.200 | TCP | webcache > 40494 [ACK] Seq=0 Ack=1119 Win=0 Len=0 |

Figure 7 - Dialogue HTTP entre un PC du LAN et la DMZ avec redirection de port 8080 -> 80

En analysant les en-têtes des requêtes HTTP reçues, on observe que la page demandée correspond bien à l'URL <http://194.10.10.30/>, qui est routée de façon transparente vers le port 8080.

En effet, le serveur DMZ reçoit les requêtes sur son interface réseau ayant pour adresse IP 194.10.20.2, sur le port 8080. Il analyse bien les requêtes comme étant en

provenance de l'adresse réelle du client (194.10.10.200) et non de l'adresse IP du routeur : il s'agit donc bien d'une redirection de port et pas d'une traduction d'adresses réseau (PAT et pas NAT).

Ce procédé permet de masquer les adresses réelles des ordinateurs de la DMZ tout en les laissant assurer leur service normalement : ces ordinateurs voient les paquets sans même savoir qu'ils ont été modifiés par une traduction de port (PAT). Il permet également de déléguer les services apparaissant comme hébergés sur le routeur sur des serveurs éventuellement plus puissants et ainsi plus adaptés à la gestion de ceux-ci.

Conclusion

Au cours de ce TP, nous avons mis en place une DMZ offrant un service d'adressage dynamique (serveur DHCP) et un espace d'hébergement de site web (serveur HTTP) d'une entreprise. Par la suite, nous avons sécurisé cette plateforme en mettant en place des règles de filtrage (firewalling).

Dans un cas réel de mise en place d'une DMZ avec les services cités ci-dessus, il faudrait envisager la mise en place d'un serveur DNS. De plus, afin d'éviter toute rupture de service, notamment dans le cas du site internet, une stratégie d'administration à distance devrait aussi être envisagée.