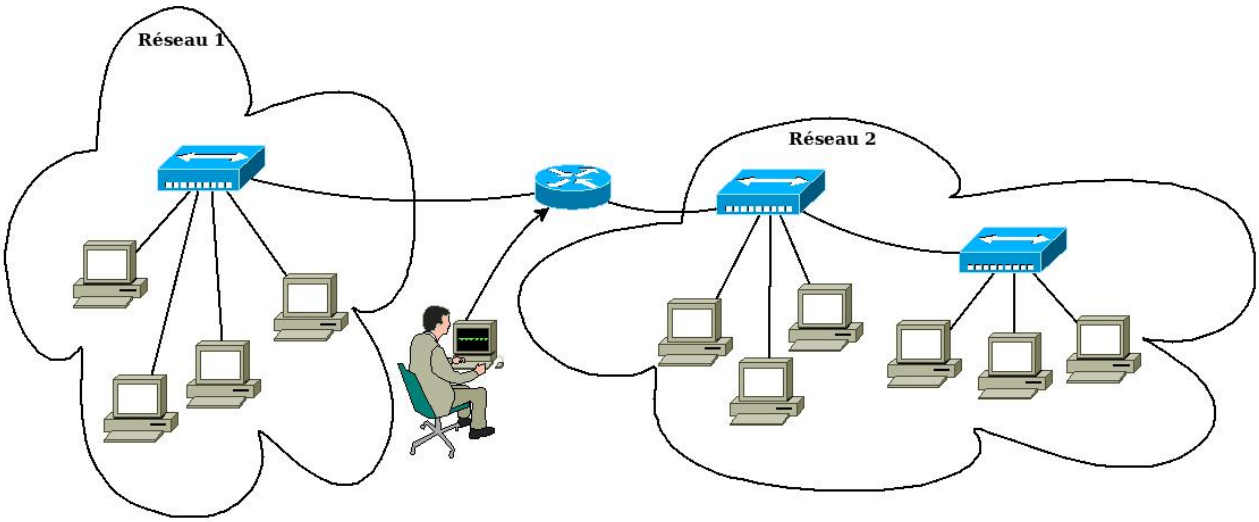
Département d'Informatique
Faculté des Sciences
Université Mohammed Premier
Oujda

Administration Réseaux



Filière : SMI
Semestre : 6

Pr. Abdelhak LAKHOUAJA
a.lakhouaja@ump.ma

http://lakhouaja.oujda-nlp-team.net/teaching/bachlor-level/administration-reseaux

Année universitaire 2015/2016

Table des matières

Bibliographie

[1] Documentation SAMBA
Français *http://doc.ubuntu-fr.org/samba*
Anglais *https://www.samba.org/samba/docs/*

[2] Site officiel de wireshark *https://www.wireshark.org*

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2009. http://www-nlp.stanford.edu/IR-book/

[4] Cédric Pruski. *Une approche adaptative pour la recherche d'information sur le Web*. Theses, Université Paris Sud - Paris XI ; université du Luxembourg, April 2009.

[5] Ian H. Witten, Marco Gori, and Teresa Numerico. *Web Dragons : Inside the Myths of Search Engine Technology*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[6] *A Little History of the World Wide Web*. http://www.w3.org/History.html

[7] *comScore Releases November 2014 U.S. Desktop Search Engine Rankings*. http://www.comscore.com/Insights/Market-Rankings/comScore-Releases-November-2014-U.S.-Desktop-Search-Engine-Rankings

Chapitre 1

Introduction

1.1 Outils systèmes et réseaux « de base »

- **ifconfig/ipconfig** : pour configurer une interface réseau ou afficher les informations concernant les interfaces réseaux (déjà utilisés en S5).
- **route :** pour afficher/modifier la table de routage.
- **ping :** pour tester la connexion entre deux machines.
- **host :** c'est un utilitaire simple de conversion de noms DNS en adresse IP (et vice versa) : **Exemple :**

```
alkhalil: $ host www.ump.ma
www.ump.ma has address 196.200.156.5
```

- **nslookup :** disponible sous Linux et sous Windows, comme la commande **host**, cette commande permet d'afficher l'adresse IP d'un nom DNS. **Exemple :**

```
alkhalil:~$ nslookup www.ump.ma
Server: 127.0.1.1
Address: 127.0.1.1#53

Non-authoritative answer:
Name: www.ump.ma
Address: 196.200.156.5
```

- **netstat**
- **tcpdump/wireshark**

1.2 Netstat

netstat -s | -a | -r | -n

Netstat fournit des statistiques sur les :

- paquets émis ou reçus
- erreurs
- collisions
- protocoles utilisés

– le nom et l'état des interfaces du système

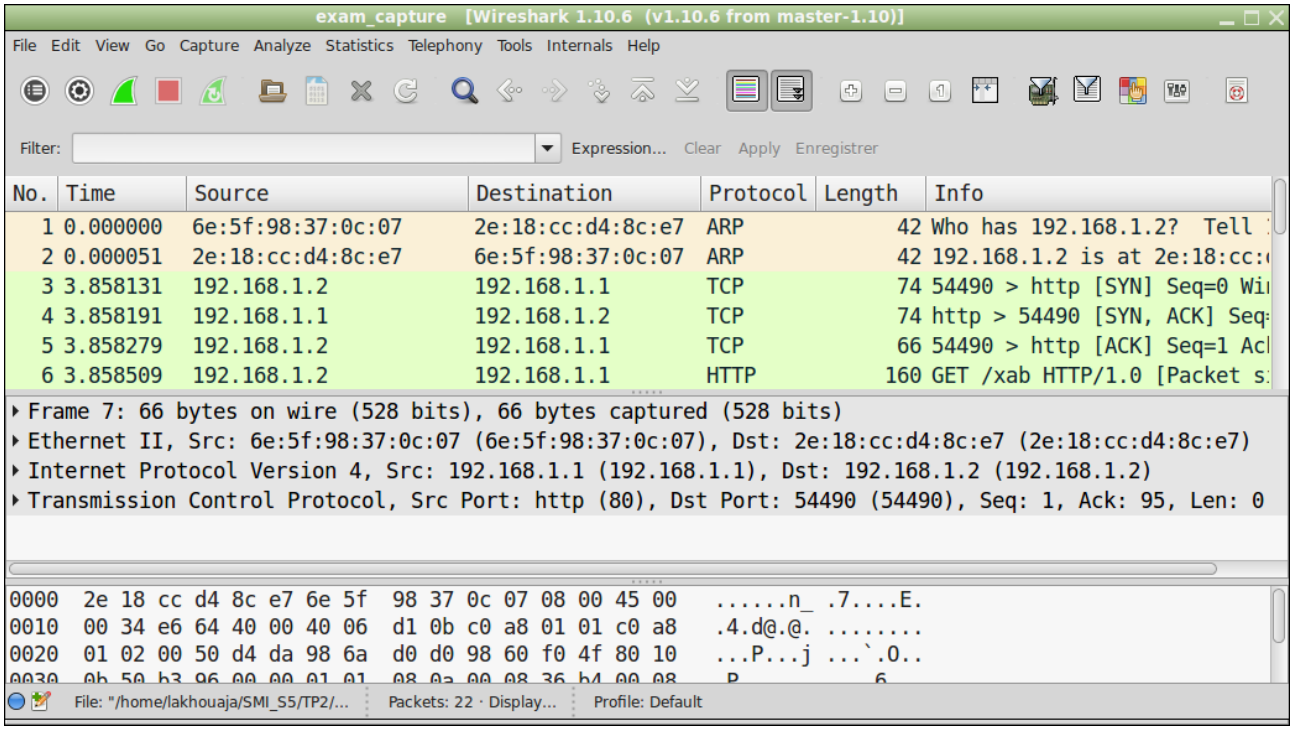
```
netstat -i
```

– le contenu de la table de routage
  `netstat -r | n`
– ainsi que l'état de tous les sockets
  `netstat -a`

1.3 Wireshark

Wireshark [1] est un outil d'analyse des réseaux qui permet de capturer et d'analyser les paquets qui circulent sur le réseau. Il peut être utilisé pour capturé les paquets qui circulent sur une interface ou pour visualiser le contenu d'un fichier qui contient des paquets capturés par un autre utilitaire tel que **tcpdump**. Il est multi-plateforme, il fonctionne sous Linux, Windows, MacOS, ...

Son interface se présente comme suit :



Les colonnes se présentent comme suit :

No. : représente le numéro du paquet ;

Time : représente le temps de capture du paquet ;

Source : représente l'adresse IP ou MAC de la source ;

Destination : représente l'adresse IP ou MAC destination ;

Protocol : représente le type du protocole capturé ;

Length : représente la taille du paquet (en octets) ;

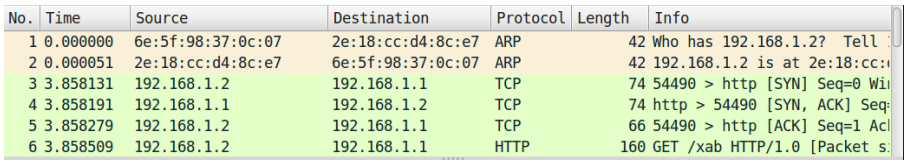
Info : représente une brève information concernant le paquet.

Remarque : sous Linux, comme pour la commande **tcpdumd**, **wireshark** ne peut pas être utilisé pour capturé des données en mode simple utilisateur. Pour capturer des données il faut passer en mode administrateur.

1. Site officiel : https://www.wireshark.org/

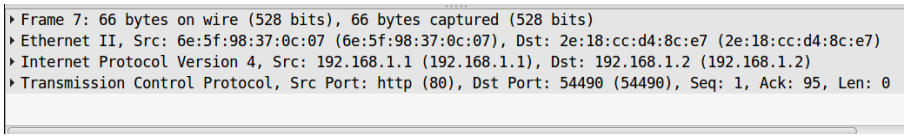
L'interface est découpé en trois zones :

1. Zone supérieure : contient l'ensemble des paquets capturés (figure suivante :)



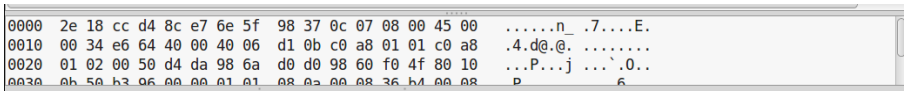
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6e:5f:98:37:0c:07	2e:18:cc:d4:8c:e7	ARP	42	Who has 192.168.1.2? Tell
2	0.000051	2e:18:cc:d4:8c:e7	6e:5f:98:37:0c:07	ARP	42	192.168.1.2 is at 2e:18:cc:
3	3.858131	192.168.1.2	192.168.1.1	TCP	74	54490 > http [SYN] Seq=0 Wi
4	3.858191	192.168.1.1	192.168.1.2	TCP	74	http > 54490 [SYN, ACK] Seq
5	3.858279	192.168.1.2	192.168.1.1	TCP	66	54490 > http [ACK] Seq=1 Ac
6	3.858509	192.168.1.2	192.168.1.1	HTTP	160	GET /xab HTTP/1.0 [Packet s

2. Zone centrale : affiche les détails d'un paquet sélectionné sous forme de couches (figure suivante :)



```
▸ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07), Dst: 2e:18:cc:d4:8c:e7 (2e:18:cc:d4:8c:e7)
▸ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
▸ Transmission Control Protocol, Src Port: http (80), Dst Port: 54490 (54490), Seq: 1, Ack: 95, Len: 0
```

3. Zone inférieure : présente le paquet sous format octale et ASCII (figure suivante :)



```
0000  2e 18 cc d4 8c e7 6e 5f  98 37 0c 07 08 00 45 00   ......n_ .7....E.
0010  00 34 e6 64 40 00 40 06  d1 0b c0 a8 01 01 c0 a8   .4.d@.@. ........
0020  01 02 00 50 d4 da 98 6a  d0 d0 98 60 f0 4f 80 10   ...P...j ...`.O..
0030  0b 50 b3 96 00 00 01 01  08 0a 00 08 36 b4 00 08    P          6
```

Filtres

Il est possible de ne pas afficher tous les paquets on les filtrant. Par exemple, on peut afficher juste les paquets **http**, en tapant **http** dans la zone **Filter :**. Il est possible aussi d'utiliser des expressions.

Exemples :

Filtre/expression	**Signification**
tcp	afficher seulement les paquets TCP
ip.src==192.168.1.2	afficher seulement les paquets qui sortent de 192.168.1.2
ip.dst==192.168.1.1 && http	afficher les paquets HTTP qui partent vers 192.168.1.2
ip && !udp	afficher les paquets IP mais n'afficher pas les paquets UDP

Chapitre 2

Serveurs web

2.1 Protocole HTTP

Le protocole HTTP (HyperText Transfer Protocol) est le protocole de transport de données le plus utilisé sur Internet depuis 1990. La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML. La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message.

Le but du protocole HTTP est de permettre un transfert de fichiers entre un navigateur (le client) et un serveur Web.

2.1.1 Pourquoi utiliser HTTP ?

HTTP est devenu le protocole de communication de l'Internet. Il :
- est disponible sur toutes les plates-formes ;
- est simple. Ne requière que peu de support pour fonctionner correctement ;
- offre un niveau de sécurité simple et efficace ;
- est utilisable à travers des pare-feu.

2.1.2 Fonctionnement

HTTP fonctionne selon le schéma classique client/serveur :
- connexion du client vers le serveur ;
- demande d'une information via une **méthode** ;
- renvoi de l'**information** ou une **erreur** ;
- déconnexion.

2.1.3 Code de retour

1xx : Information

2xx : Succès (par exemple : 200 ok).

3xx : Redirection.

4xx : Erreurs (par exemple : 404 Not Found).

5xx : Erreurs venant du serveur HTTP (par exemple : 501 Not Implemented).

On verra en TP plus de codes de retours.

2.1.4 Méthodes

Les méthodes HTTP sont les suivantes :

GET : demande de la ressource située à l'URL spécifiée ;

HEAD : demande de l'en-tête de la ressource située à l'URL spécifiée ;

POST : envoi de données au programme situé à l'URL spécifiée ;

PUT : envoi de données à l'URL spécifiée ;

DELETE : suppression de la ressource située à l'URL spécifiée.

2.1.5 Requête HTTP

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- une ligne de requête précise la méthode qui doit être appliquée, et la version du protocole utilisée. La ligne comprend trois éléments devant être séparés par un espace :
  - la méthode ;
  - l'URL ;
  - la version du protocole utilisé par le client (généralement HTTP/1.0) ;

  Exemple :
  GET / HTTP/1.0
- les champs d'en-tête de la requête : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...).

2.1.6 Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace :
  - La version du protocole utilisé
  - Le code de statut
  - La signification du code
- Les champs d'en-tête de la réponse : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête
- Le corps de la réponse : il contient le document demandé

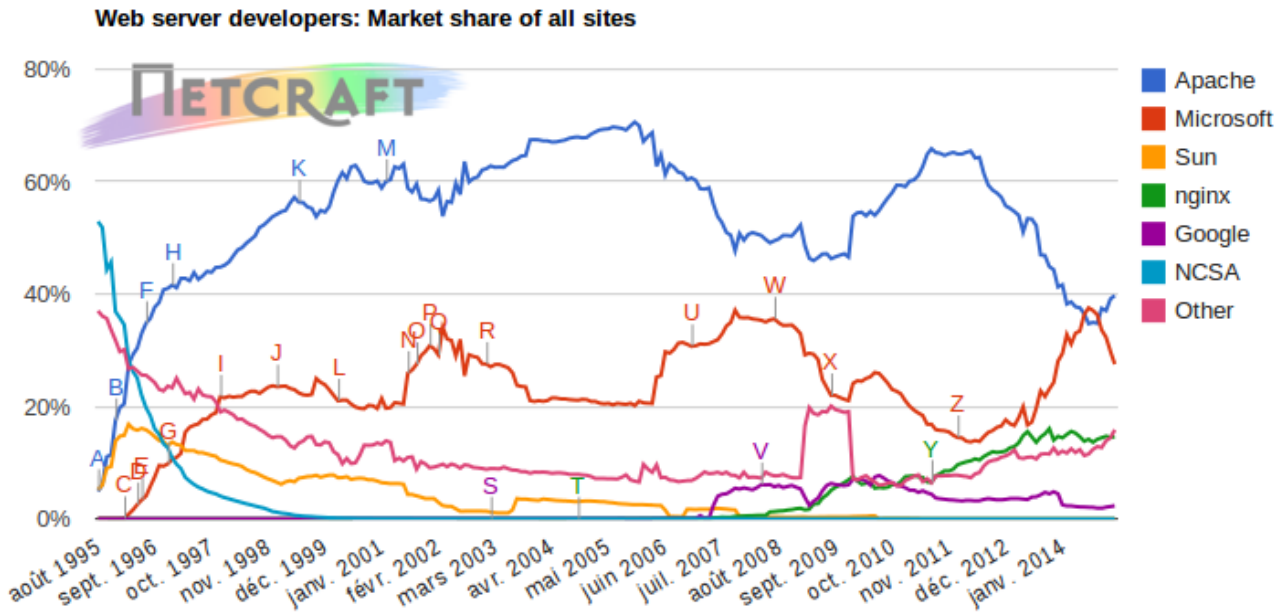
2.2 Serveurs Web

Principaux serveurs :

- Apache
- Microsoft : Internet Information Server (IIS)
- nginx
- gws (Google Web Server)

2.2.1 Serveurs Web en ligne 2015

D'après Netcraft [1], le serveur apache est toujours les serveur web le plus utilisé (voir figure suivante).



Le tableau suivant, montre la part du marché des principaux fournisseurs de serveurs web (selon netcraft).

Développeur	12/2014	Pourcentage	01/2015	Pourcentage	Changement
Apache	358 159 405	39,11%	348 460 753	39,74%	0,63
Microsoft	272 967 294	29,81%	241 276 347	27,52%	-2,29
nginx	132 467 763	14,47%	128 083 920	14,61%	0,14
Google	20 011 260	2,19%	20 209 649	2,30%	0,12

1. http://survey.netcraft.com

2.3 Serveur web apache

2.3.1 Généralités

Comme on l'a vu dans la section précédente, apache [2] est le serveur web le plus populaire sur Internet. Il est robuste et extensible. Il est distribué sous une licence "Open source" (Licence Apache).

Il est disponible sur plusieurs plateformes (Linux,windows, ...)

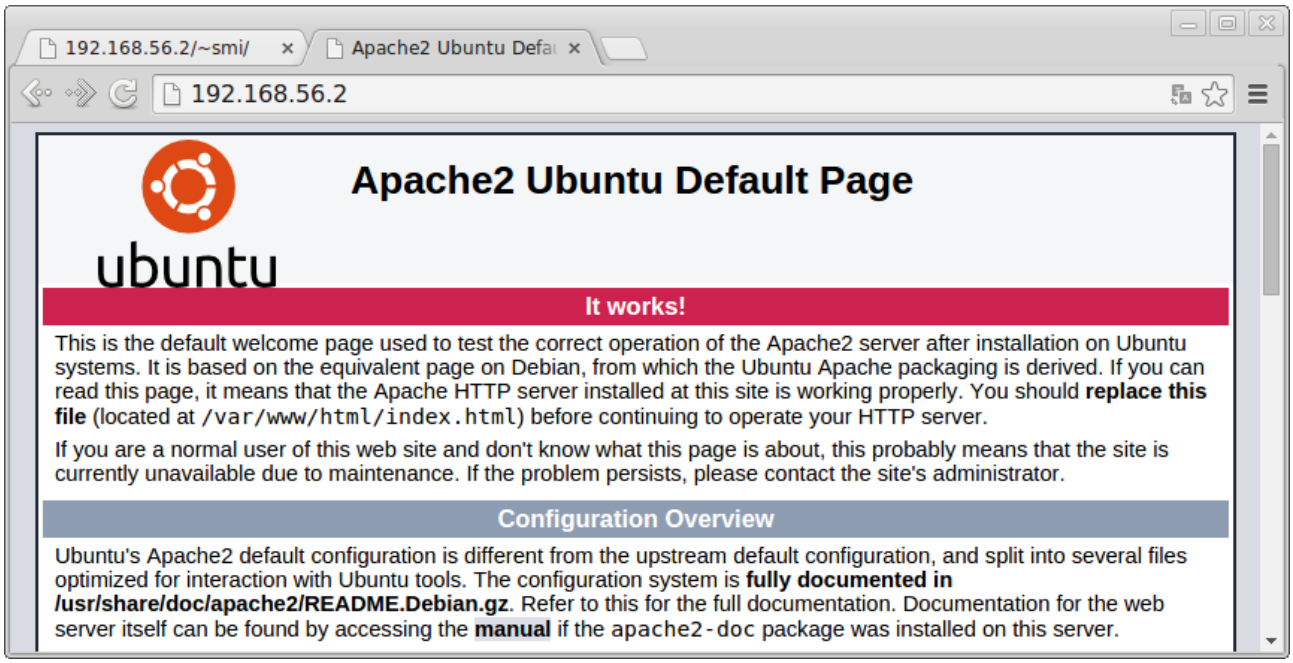
2.3.2 Installation minimale

Une installation minimale peut être faite en ligne de commande de la façon suivante :

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

2.3.3 Vérification de l'installation

Pour vérifier l'installation, il suffit d'utiliser un navigateur web.



2.3.4 Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire `/etc/apache2`.

apache2.conf : fichier de configuration principale.

envars : contient les variables d'environnement propres à apache.

conf-available/ : contient des fichiers de configuration additionnels disponibles.

2. Site officiel : http://httpd.apache.org/

conf-enabled/ : contient des fichiers de configuration activés. Il sont utilisés dans **apache2.conf** par la ligne :
`IncludeOptional conf-enabled/*.conf`

ports.conf : directives de configuration pour les ports et les adresses IP d'écoutes.

mods-available/ : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

mods-enabled/ : pour utiliser un module (activer), il faut mettre un lien symbolique vers le fichier .load (et .conf, s'il existe) du module associé dans le dossier mods-available.

sites-available/ : même chose que modsavailable/, mais cette fois pour les sites virtuels. Ce n'est pas obligé d'avoir le même nom pour le site et le fichier.

sites-enabled/ : même chose que modsenabled/.

magic : instructions pour déterminer le type **MIME** d'un fichier (**M**ultipurpose **I**nternet **M**ail **E**xtensions - Extensions Multi-usages de la Messagerie par Internet). Par exemple text/html et image/gif.

Remarque :

Par défaut, un seul serveur est disponible (le serveur par défaut). Il est disponible dans **apache2.conf** par la ligne :
`IncludeOptional sites-enabled/*.conf`

2.3.5 Activer/désactiver un module

Les commandes **a2enmod** et **a2dismod** sont disponibles pour activer ou désactiver un module.

Exemple : pages web personnelles

Pour permettre aux utilisateurs d'avoir leurs propres pages web disponibles via un lien de type :

`http://NomSite/~utilisateur`
`http://localhost/~smi`

On tape la commande

```
#a2enmod userdir
```

Il faut ensuite redémarrer apache en tapant la commande :

```
#service apache2 restart
```

Exemple d'une page personnelle :

Dans le répertoire personnelle de l'utilisateur **smi**, il faut créer le répertoire **public_html** avec les droits `-rwxr-xr-x` et mettre dedans le fichier **index.html**, avec les droits `-rwxr--r--`.

2.3.6 Activer/désactiver un site

Les commandes **a2ensite** et **a2dissite** sont disponibles pour activer ou désactiver un site. On verra leurs utilisation dans les sections suivantes.

2.3.7 Configuration de base

Avant de commencer la configuration, il faut faire une sauvegarde des fichiers que vous voulez modifier. Par exemple :

```
#cp apache2.conf apache2.conf.save
```

— Port à écouter (ports.conf) :
`Listen 80`

— Emplacement par défaut des pages html : `/var/www/html`
mettre les fichiers concernant le site web dans ce répertoire.

— Pages par défaut (mods-enabled/dir.conf) :
`DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm`

2.3.8 Sites virtuelles

Apache permet de gérer plusieurs sites web. Chaque site est appelé serveur virtuel et possède sa propre configuration.

Il y a deux types de serveurs virtuels :

— Serveurs par nom ; les mêmes sites utilisent la même adresse IP. Par exemple :

IP	**Nom**
192.168.56.2	smi.ump.ma
192.168.56.2	sma.ump.ma

— Serveurs par adresse IP ; chaque site utilise sa propre adresse IP. Par exemple :

IP	**Nom**
192.168.56.2	smi.ump.ma
192.168.56.10	sma.ump.ma

2.3.9 Configuration de deux sites virtuels par nom

Dans cette exemple, nous allons configurer deux sites virtuels, le premier **smi.ump.ma** et le deuxième **sma.ump.ma**. Les deux sites utilisent la même adresse IP **192.168.56.2**.

Il faut déclarer les deux noms dans le fichier **/etc/hosts** :

```
192.168.56.2      smi.ump.ma      www.smi.ump.ma
192.168.56.2      sma.ump.ma      www.sma.ump.ma
```

On pourra utiliser un serveur DNS pour déclarer les noms (voir chapitre concernant le DNS).

Il faut créer les répertoires **smi** et **sma** associés dans **/var/www/html** :

```
# mkdir /var/www/html/smi
# mkdir /var/www/html/sma
```

Dans **/etc/apache2/sites-available/**, il faut créer deux fichiers : **smi.conf** et **sma.conf**

Contenu du fichier smi.conf

```
<VirtualHost *:80>
    DocumentRoot /var/www/html/smi
    ServerName   smi.ump.ma
    ServerAlias www.smi.ump.ma
</VirtualHost>
```

Avec :

DocumentRoot : emplacement par défaut des pages html ;

ServerName : nom du serveur virtuel ;

ServerAlias : autre nom (alias) du serveur virtuel.

Contenu du fichier sma.conf

```
<VirtualHost *:80>
    DocumentRoot /var/www/html/sma
    ServerName   sma.ump.ma
    ServerAlias www.sma.ump.ma
</VirtualHost>
```

Activation des deux sites

Il faut activer les deux sites en tapant les commandes :

```
# a2ensite smi
# a2ensite sma
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Les deux sites seront accessibles via les liens :

http://smi.ump.ma ou http://www.smi.ump.ma

http://sma.ump.ma ou http://www.sma.ump.ma

2.3.10 Sites virtuels par adresse IP

Dans cette exemple, nous allons configurer un nouveau site virtuel **smp.ump.ma**, qui utilise une adresse IP différente.

Dans cette exemple, la machine doit être muni, soit de plusieurs interfaces réseaux soit de plusieurs adresses IP associées à la même interface réseau (on parle d'IP aliasing).

IP aliasing

Pour affecter une seconde adresse IP à une interface réseau, il faut exécuter la commande :

```
# ifconfig eth0:0 192.168.56.10 up
```

Remplacez eth0 par une autre interface (par exemple eth1).

L'interface dispose, maintenant, de deux adresses distinctes :
- Adresse IP :192.168.56.2
- Alias IP : 192.168.56.10

A vérifier avec la commande :

```
# ifconfig
```

Pour rendre cette configuration permanente, il faut ajouter les lignes suivantes au fichier **/etc/network/interfaces** :

```
auto eth0:0
iface eth0:0 inet static
address 192.168.56.10
netmask 255.255.255.0
```

Remarque : on peut ajouter autant d'interfaces qu'on veut (eth0:1, eth0:2 ...).

Configuration du site virtuel

Il faut ajouter au fichier **/etc/hosts**, les lignes suivantes :

```
192.168.56.10          smp.ump.ma
192.168.56.10          www.smp.ump.ma
```

Il faut créer le répertoire **smp** associé dans /**var**/**www**/**html** :

```
# mkdir /var/www/html/smp
```

Dans /**etc**/**apache2**/**sites-available**/, il faut créer le fichier : **smp.conf**

Contenu du fichier smp.conf

```
<VirtualHost 192.168.56.10:80>
    DocumentRoot /var/www/html/smp
    ServerName smp.ump.ma
    ServerAlias www.smp.ump.ma
</VirtualHost>
```

Activation du nouveau site

Il faut activer le site en tapant la commande :

```
# a2ensite smp
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Le nouveau site sera accessible via les liens :

http://smp.ump.ma ou http://www.smp.ump.ma

2.4 Sécuriser apache

Apache est très modulaire. Dans le chapitre suivant, on verra un module concernant **php**. Dans cette section, on va utiliser un module important dans l'aspect sécurité. Le module **mod_ssl** ajoute la possibilité de crypter les communications entre le client et le serveur.

Le mode **mod_ssl** se trouve dans le package **apache2-common**. Pour l'activer, il faut taper la commande :

```
sudo a2enmod ssl
```

suivie de la commande

```
sudo service apache2 restart
```

Après l'activation, il faut utiliser le préfixe **https ://** devant l'adresse du serveur dans la barre du navigateur (par exemple : https://192.168.56.2/).

Remarque : dans le chapitre précédent, on a vu que le site officiel de **wireshark** (https://www.wireshark.org) utilise une connexion sécurisé.

Chapitre 3

Php et MySQL

3.1 PHP

PHP est un Langage de script interprété (non compilé) spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML [1].

Pour installer la version 5 de PHP, il faut exécuter la commande :

```
# sudo apt-get install php5 libapache2-mod-php5
```

Il faut ensuite redémarrer apache :

```
# sudo service apache2 restart
```

3.1.1 Vérification de l'installation

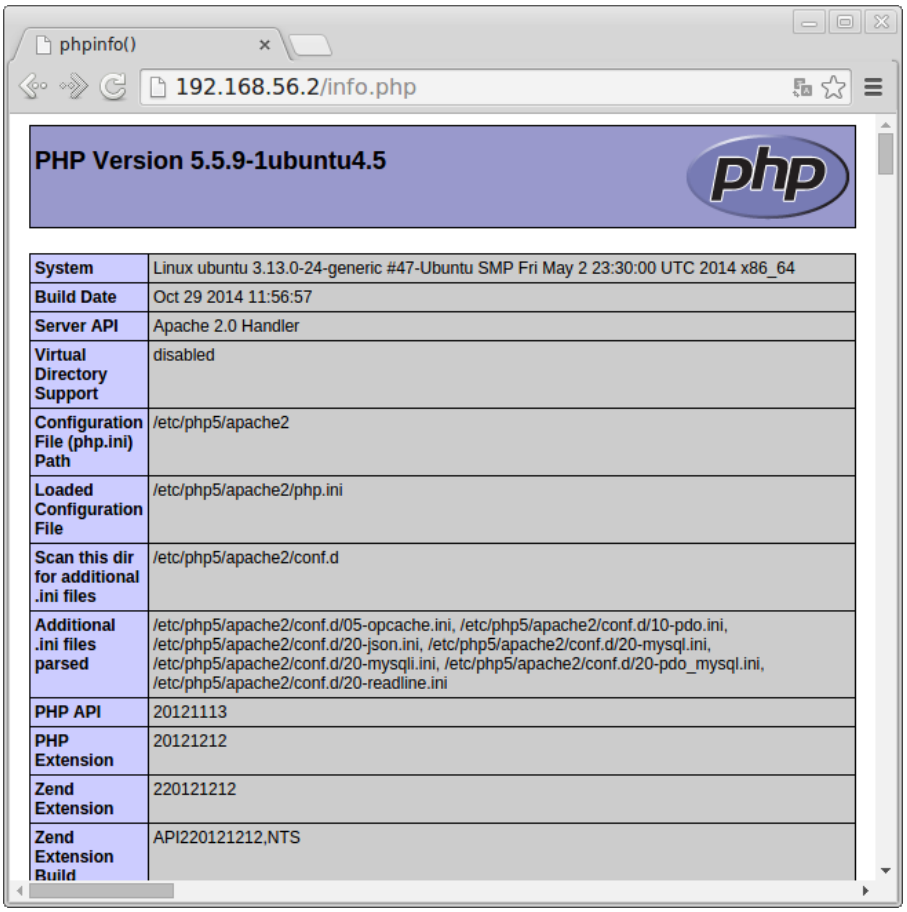
Dans le répertoire `/var/www/html`, créez le script **info.php** :

```
<?php
    phpinfo();
?>
```

Dans votre navigateur, tapez l'adresse `http://localhost/info.php` ; remplacez **localhost** par l'adresse du serveur, par exemple 192.168.56.2, il fournira un ensemble d'informations et de paramètres de configuration.

Dans le figure suivante, nous donnons le résultat de l'exécution du script **info.php**.

1. voir `php.net`



3.1.2 Utilisation

L'utilisation de **php** sort du cadre de ce cours, il concerne le cours « Technologie du web ». La documentation complète est disponible sur le site `php.net`. Il existe aussi plusieurs livres concernant **php**.

3.2 MySQL

MySQL est un système de gestion de bases de données relationnel (SGBDR) libre, open-source et gratuit. Il est performant et très populaire. Il est multi-utilisateurs.

3.2.1 Installation

Pour l'installer, il faut taper la commande :

```
# sudo apt-get install mysql-server
```

Durant l'installation, vous devez saisir le mot de passe de l'administrateur (**root**) de MySQL. Il a le même nom que l'administrateur Linux (à ne pas confondre les noms !)

3.2.2 Documentation

Pour plus d'informations sur MySQL, veuillez consulter les sites : http://www.mysql.com/ et http://dev.mysql.com/doc/. Il existe aussi plusieurs livres concernant l'utilisation de MySQL. Vous pouvez appliquer ce que vous avez vu dans le cours de « Bases de données ».

3.2.3 Oubli du mot de passe

Si vous avez oublier le mot de passe de **root** de MySQL, vous pouvez établir un nouveau mot passe en tapant la commande :

```
sudo dpkg-reconfigure mysql-server-5.5
```

Le démon MySQL sera arrêté et vous devez saisir un nouveau mot de passe. Après la saisie, le démon MySQL sera de nouveau démarré.

3.2.4 Utilisation

Dans une console, tapez la commande :

```
mysql -u root -p
```

et tapez votre mot de passe.

Vous arriverez alors sur un prompt de type :

```
mysql>
```

Vous pouvez alors taper des requêtes MySQL. N'oubliez pas le point-virgule à la fin de la requête.

Par exemple, pour créer une base de données qui s'appelle **smi**, tapez la requête :

```
mysql> create database smi;
```

Pour voir les bases de données, tapez la requête :

```
mysql> show databases;
```

3.3 Utilisation de MySQL avec Php

Pour utiliser MySQL avec Php, il faut installer le paquet **php5-mysql**. Pour installer **php5-mysql**, il faut taper la commande :

```
sudo apt-get install php5-mysql
```

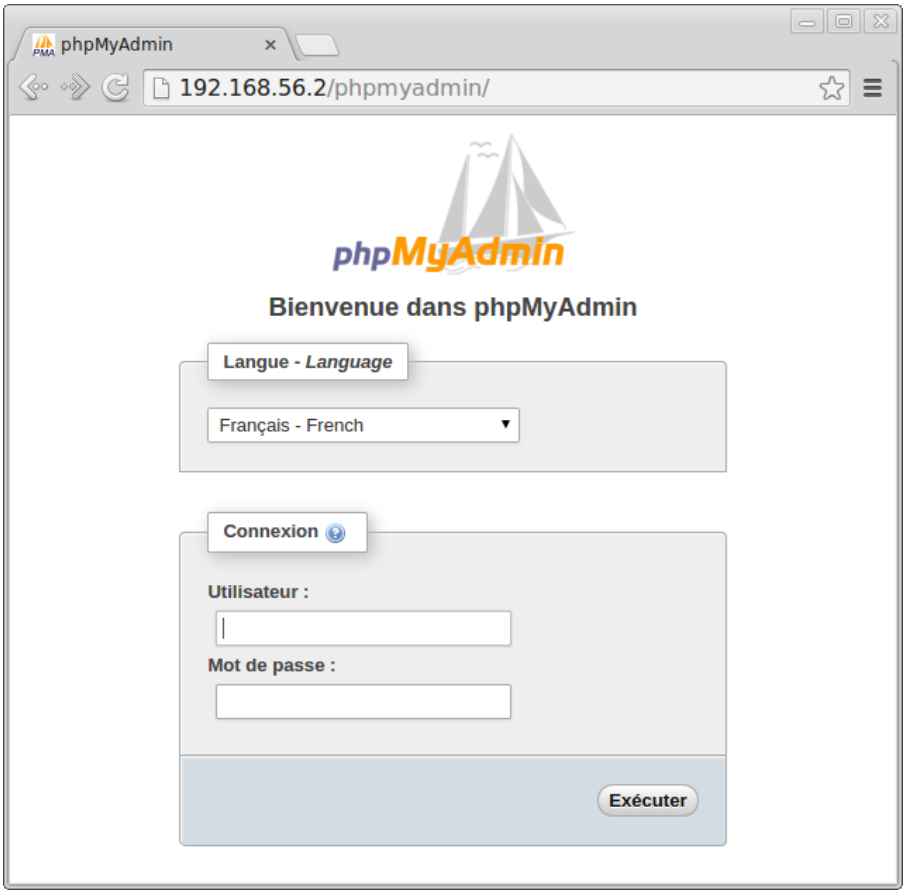
Après l'installation de **php5-mysql**, vous pouvez utiliser des applications web qui utilisent **php** comme langage de programmation et peuvent accéder à **MySQL**. Dans la section suivante, nous allons voir **phpmyadmin** qui est une application web écrite en **php** et se connecte à **MySQL**.

3.4 PhpMyAdmin

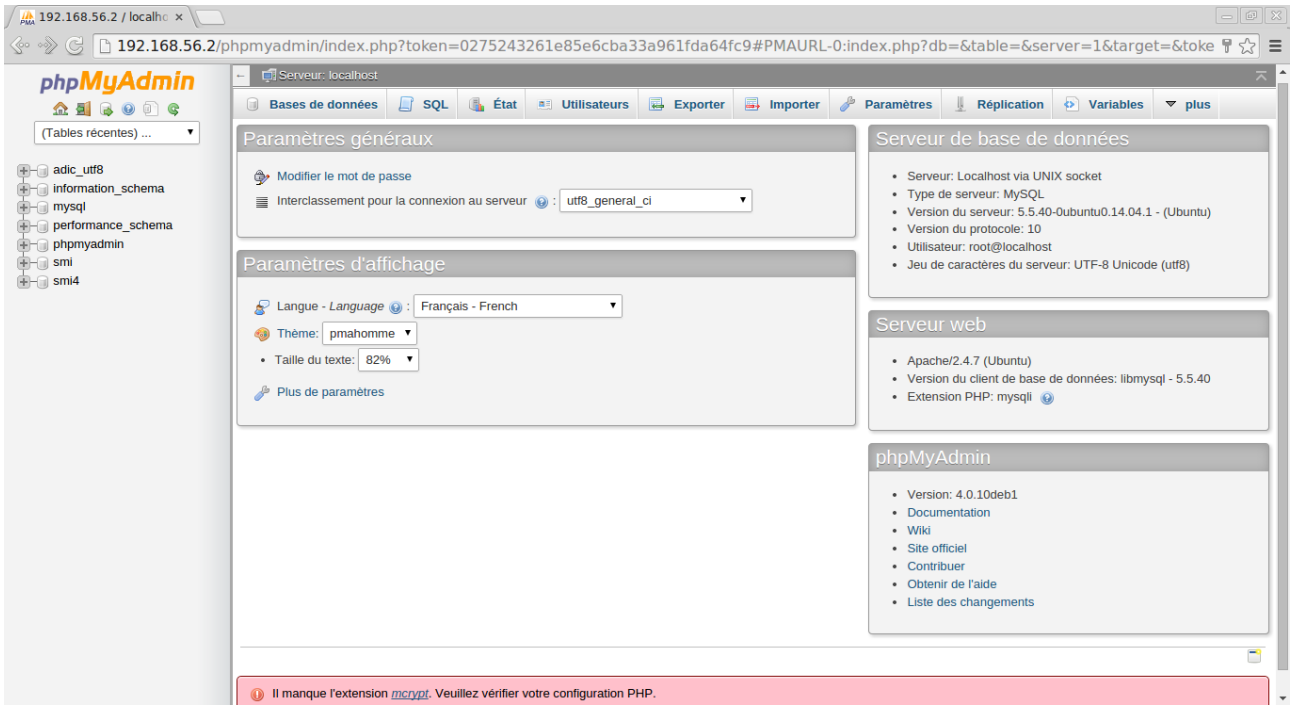
phpMyAdmin est une application écrite en PHP très utile pour l'administration de MySQL. Elle est accessible via un navigateur. Pour l'installer, tapez la commande :

```
sudo apt-get install phpmyadmin
```

Pour son utilisation, saisissez l'adresse `http://localhost/phpmyadmin`. Vous pouvez remplacer **localhost** par l'adresse de votre serveur. Dans le figure suivante, nous donnons la page de connexion de **phpmyadmin**.



Dans le figure suivante, nous donnons la page d'accueil de **phpmyadmin**.


192.168.56.2 / localhc
192.168.56.2/phpmyadmin/index.php?token=0275243261e85e6cba33a961fda64fc9#PMAURL-0:index.php?db=&table=&server=1&target=&toke
phpMyAdmin
(Tables récentes) ...
adic_utf8
information_schema
mysql
performance_schema
phpmyadmin
smi
smi4
Serveur: localhost
Bases de données
SQL
État
Utilisateurs
Exporter
Importer
Paramètres
Réplication
Variables
plus
Paramètres généraux
Modifier le mot de passe
Interclassement pour la connexion au serveur : utf8_general_ci
Paramètres d'affichage
Langue - Language : Français - French
Thème: pmahomme
Taille du texte: 82%
Plus de paramètres
Serveur de base de données
Serveur: Localhost via UNIX socket
Type de serveur: MySQL
Version du serveur: 5.5.40-0ubuntu0.14.04.1 - (Ubuntu)
Version du protocole: 10
Utilisateur: root@localhost
Jeu de caractères du serveur: UTF-8 Unicode (utf8)
Serveur web
Apache/2.4.7 (Ubuntu)
Version du client de base de données: libmysql - 5.5.40
Extension PHP: mysqli
phpMyAdmin
Version: 4.0.10deb1
Documentation
Wiki
Site officiel
Contribuer
Obtenir de l'aide
Liste des changements
Il manque l'extension mcrypt. Veuillez vérifier votre configuration PHP.

Chapitre 4

Serveur DHCP

4.1 Introduction

Une adresse réseau peut être configurer soit de manière statique ou dynamique :

Statique : l'utilisateur configure lui même l'adresse IP de la machine.

Dynamique : la machine obtient l'adresse grâce à un serveur DHCP.

Le serveur DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) est un protocole de configuration dynamique de machines, il permet l'affectation, de façon automatique, des paramètres réseaux à une machine.

En général, le serveur DHCP affecte à un client :
- l'adresse IP ;
- la passerelle par défaut ;
- les adresses IP des serveurs DNS.

Le serveur DHCP peut affecter aussi :
- le nom de la machine ;
- le nom du domaine ;
- le serveur d'impression ;
- le serveur de temps (qui donne le temps à la machine).

Le serveur DHCP attribue les paramètres suivant deux méthodes :

automatique : pour une période de temps, il affecte une adresse IP à partir d'un intervalle au client. Si le client n'est pas connecté pour une certaine période de temps, l'adresse peut être affecté à une autre machine ;

fixe : en utilisant l'adresse MAC d'une machine, le serveur DHCP affecte toujours la même adresse IP à la machine. Ceci pour assurer qu'une machine avec une cette adresse MAC, reçoive toujours la même adresse IP.

Remarque : il ne faut pas confondre statique et fixe. Statique veut dire que c'est l'utilisateur qui configure l'adresse IP de sa machine.

L'avantage de l'utilisation d'un serveur DHCP est que toute changement dans les paramètres réseaux se fera au niveau du serveur DHCP. Un autre avantage est la facilité d'ajout de nouvelles machines dans e réseau.

On peut avoir des serveurs DHCP sous Linux et sous windows-server. Dans ce qui suit, nous allons utiliser le serveur **isc-dhcp-server**.

4.2 Installation du serveur isc-dhcp-server

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

Vous devez changer la configuration par défaut, en modifiant les deux fichiers **/etc/dhcp/dhcpd.conf** et **/etc/default/isc-dhcp-server**.

4.3 Configuration

Deux cas seront traités :

- adresse fixe alloué à la machine web-smi ;
- adresses dynamiques alloués aux autres machines.

On suppose que le serveur dispose de trois interfaces réseaux :

- eth0 : interface pour se connecter à Internet ; adresse obtenue par dhcp à partir d'un autre serveur DHCP ;
- eth1 dont l'adresse IP est : 192.168.1.1 ;
- eth2 dont l'adresse IP est : 192.168.10.1 ;

4.3.1 Interface(s) d'écoute(s)

Si vous voulez que le serveur écoute sur certaines interfaces vous devez les spécifier dans **/etc/default/isc-dhcp-server**. Dans notre cas, le fichier doit contenir la ligne :

```
INTERFACES="eth1 eth2"
```

L'écoute se fera sur les interfaces eth1 et eth2.

4.3.2 Configuration du serveur

La configuration se fait dans le fichier **/etc/dhcp/dhcpd.conf**.

Les options sont définies de façon globale ou par réseau. Dans ce qui suit, nous allons voir un exemple de configuration pour le cas traité.

Options générales

Dans l'exemple suivant, on donnera les options communes aux différents reseaux.

```
# Nom du domaine DNS
option domain-name "ump.ma";

# Nom(s) de(s) serveur(s) DNS
option domain-name-servers 192.168.100.10, 192.168.10.11;

# Temps de renouvellement des adresses en s (1h)
default-lease-time 3600;
```

```
# maximum (2h)
max-lease-time 7200;

# Mode autoritaire
# Est-ce-que ce serveur DHCP est le serveur principal?
authoritative;

# Masque de sous-reseau
option subnet-mask 255.255.255.0;
```

Configuration du réseau 192.168.1.0

```
# declaration du sous reseau 192.168.1.*
subnet 192.168.1.0 netmask 255.255.255.0 {
    # Adresse de diffusion
    option broadcast-address 192.168.1.255;

    # routeur par defaut
    option routers 192.168.1.1;

    # intervalle des adresses
    range 192.168.1.2 192.168.1.100;
}
```

Configuration du réseau 192.168.10.0

```
# declaration du sous reseau 192.168.10.*
subnet 192.168.10.0 netmask 255.255.255.0 {
    # specifier un domaine different de celui par defaut :
    option domain-name "fso.ump.ma";

    # Adresse de diffusion
    option broadcast-address 192.168.10.255;

    # routeur par defaut
    option routers 192.168.10.1;

    # intervalle des adresses
    range 192.168.10.20 192.168.10.200;
}
```

Configuration de la machine « web-smi »

```
host web-smi {
    # adresse mac de la carte reseau
    # A remplacer par celle de la machine
    hardware ethernet 08:00:27:A6:C2:50;

    # adresse attribue
    fixed-address 192.168.1.200;
}
```

Remarque : si le réseau 192.168.1.0, ne figure pas dans le fichier de configuration, il faut le signaler de la façon suivante :

```
# Ajouter pour comprendre a topologie du reseau
# Ne fourni aucun service
subnet 192.168.1.0 netmask 255.255.255.0 {
}
```

4.3.3 Redémarrage du serveur DHCP

Après avoir changé les fichiers de configuration, il faut redémarrer le démon **dhcpd** :

`sudo service isc-dhcp-server restart`

4.4 Fonctionnement de DHCP

La figure suivante, présente une visualisation par wireshark d'une capture de paquet lors de l'affectation d'une adresse à un client DHCP.

No.	Time	Source	Destination	Protocol	Length	Info
3	1.059241	0.0.0.0	255.255.255.255	BOOTP	342	Boot Request from 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07)[Packet
4	1.059960	192.168.1.1	192.168.1.2	ICMP	62	Echo (ping) request id=0xc8c1, seq=0/0, ttl=64
5	1.405486	192.168.1.1	192.168.1.2	BOOTP	342	Boot Reply[Packet size limited during capture]
6	1.406153	0.0.0.0	255.255.255.255	BOOTP	342	Boot Request from 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07)[Packet
7	1.406686	192.168.1.1	192.168.1.2	BOOTP	342	Boot Reply[Packet size limited during capture]
11	6.054863	0e:e5:85:cc:fb:38	6e:5f:98:37:0c:07	ARP	42	Who has 192.168.1.2? Tell 192.168.1.1
12	6.055121	6e:5f:98:37:0c:07	0e:e5:85:cc:fb:38	ARP	42	192.168.1.2 is at 6e:5f:98:37:0c:07

Source	Destination	Protocol	Info
0.0.0.0	255.255.255.255	DHCP	DHCP Discover (utilise UDP) Le client utilise l'adresse 0.0.0.0 (hôte inconnu) et envoi la demande à toutes les machines du réseau.
192.168.1.1	192.168.1.2	ICMP	Echo (ping) request Avant d'affecter l'adresse 192.168.1.2 au client, le serveur DHCP s'assure que cette adresse n'est pas utilisé par une autre machine.

Source	Destination	Protocol	Info
192.168.1.1	192.168.1.2	DHCP	DHCP Offer Le serveur DHCP offre l'adresse 192.168.1.2 au client.
0.0.0.0	255.255.255.255	DHCP	DHCP Request Le client demande l'adresse.
192.168.1.1	192.168.1.2	DHCP	DHCP ACK (acknowledgment - acquittement) Le serveur envoi un accusé d'acceptation.
0e:e5:85:cc:fb:38	6e:5f:98:37:0c:07	ARP	Who has 192.168.1.2? Tell 192.168.1.1 Demande ARP de la part du serveur
6e:5f:98:37:0c:07	0e:e5:85:cc:fb:38	ARP	192.168.1.2 is at 6e:5f:98:37:0c:07 Réponse ARP

Chapitre 5

ftp et ssh

5.1 Introduction

FTP (File Transfer Protocol - Protocle pour le Transfert de Fichiers) est un protocole TCP qui permet le téléchargement de fichiers à partir d'un serveur. Ce protocole n'est pas sécurisé du que l'envoie des données entre le client et le serveur n'est pas crypté. Pour l'opération inverse (chargement) et pour plus de sécurité, on peut utiliser **ssh** (Secure shell).

5.2 ftp

Il existe plusieurs serveurs ftp, **tftpd**, **proftp**, **twoftpd**, ...

ftp permet l'accès de deux façons :

- anonyme : l'accès se fera au serveur via le nom d'utilisateur par défaut « anonymous » ou « ftp » ;
- authentifié : l'utilisateur doit disposé sur le système distant d'un compte. Cette est déconseillé du fait que la connexion au serveur n'est pas sécurisée.

Dans cette section, nous allons utiliser **vsftpd** qui est facile à installer et à maintenir. Pour l'installer, tapez la commande :

```
sudo apt-get install vsftpd
```

5.2.1 Configuration de la connexion anonyme

Par défaut, **vsftpd** n'est pas configuré pour autorisé la connexion anonyme. Pour l'autorisée, modifiez le fichier **/etc/vsftpd.conf** en changeant la ligne :

anonymous_enable=YES

Par défaut, la valeur était **NO**. Après cette modification, il faut redémarrer le serveur ftp en tapant la commande :

```
sudo restart vsftpd
```

Durant l'installation, l'utilisateur **ftp** avec le répertoire personnel **/srv/ftp** seront créés. Les fichiers qui seront visibles par connexion ftp anonyme doivent être mises dans ce répertoire.

La commande :

```
tail -n1 /etc/passwd
```

Fournira le résultat :

```
ftp:x:111:119:ftp daemon,,,:/srv/ftp:/bin/false
```

5.2.2 Configuration de la connexion authentifiée

Par défaut, **vsftpd** est configuré pour autorisé les utilisateurs authentifiés à télécharger des fichiers. Il n'autorise ni le chargement de fichiers ni la création de répertoires. Pour autoriser le chargement de fichiers et la création de répertoires, il faut éditer le fichier **/etc/vsftpd.conf** et enlever le commentaire à la ligne :

#write_enable=YES

pour devenir :

write_enable=YES

Après, il faut redémarrer le serveur **vsftpd** en tapant la commande :

```
sudo restart vsftpd
```

Remarque

Pour d'autres options, consulter le manuel du fichier **vsftpd.conf** :

```
man vsftpd.conf
```

5.2.3 Connexion à partir d'un client

On peut se connecter à un serveur ftp, soit à partir d'un :

- terminal ;
- navigateur.

Connexion à partir d'un terminal

Pour se connecter à partir d'un terminal, il faut taper la commande :

```
ftp 192.168.56.2
```

Changez **192.168.56.2** par l'adresse ou le nom de votre serveur.

En validant la commande, vous obtiendrez :

```
Connected to 192.168.56.2.
220 (vsFTPd 3.0.2)
Name (192.168.56.2:lakhouaja): ftp
331 Please specify the password.
Password:
230 Login successful.
```

```
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Pour fermer la connexion, tapez **quit** ou **exit** dans l'invite de commandes de ftp. Pour plus de commandes, tapez dans l'invite « help » ou « ? ». Pour l'aide sur une commande, tapez : ? commande (par exemple : `ftp> ? get`). Vous pouvez aussi utiliser le manuel en ligne de ftp :

`man ftp`.

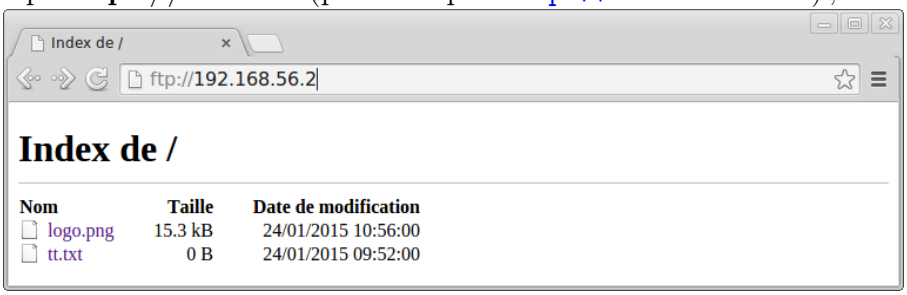
Pour une connexion :

- anonyme, tapez **ftp** ou **anonymous** après : de Name ; pour le mot de passe, il faut juste valider par la touche « Entrée » ;
- authentifié, tapez votre nom de connexion après : de Name et saisissez votre mot de passe.

Connexion à partir d'un navigateur

Vous pouvez vous connecter au serveur ftp en utilisant un navigateur.Pour une connexion :

- anonyme, tapez **ftp ://adresse** (par exemple : `ftp://192.168.56.2`) ;



- authentifié, tapez `ftp://login@adresse` (par exemple : `ftp://smi@192.168.56.2`) après validation, saisissez votre mot de passe.

5.3 ssh

Comme nous l'avons signalé au début de ce chapitre, pour le transfert de fichiers en utilisant une connexion sécurisée, il faut utiliser **ssh** (Secure Shell). Sous Linux, le serveur **ssh** disponible de façon libre et gratuite s'appelle **OpenSSH**.

5.3.1 Installation

Pour installer le client, tapez la commande

`sudo apt-get install openssh-client`

Pour installer le serveur, tapez la commande :

`sudo apt-get install openssh-server`

5.3.2 Connexion à partir d'un client Linux

Pour se connecter à partir d'un client, tapez : `ssh login@adresse`. Par exemple :

```
ssh smi@192.168.56.2
```

Pour utiliser le serveur ssh comme serveur ftp sécurisé, tapez la commande : `sftp login@adresse`. Par exemple :

```
sftp smi@192.168.56.2
```

Après saisi du mot de passe, vous obtiendrez l'invite de commandes :

```
sftp>
```

Pour fermer la connexion, tapez **quit**, **bye** ou **exit** dans l'invite de commandes. Pour plus de commandes, tapez dans l'invite « help » ou « ? ». Vous pouvez aussi utiliser le manuel en ligne de sftp :

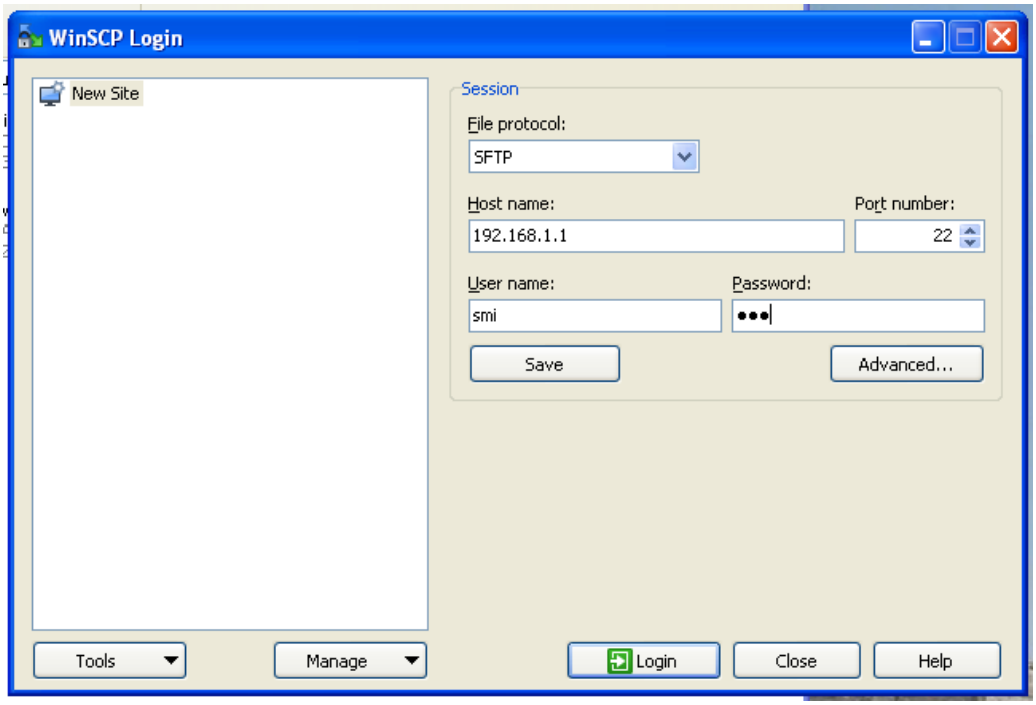
`man sftp`.

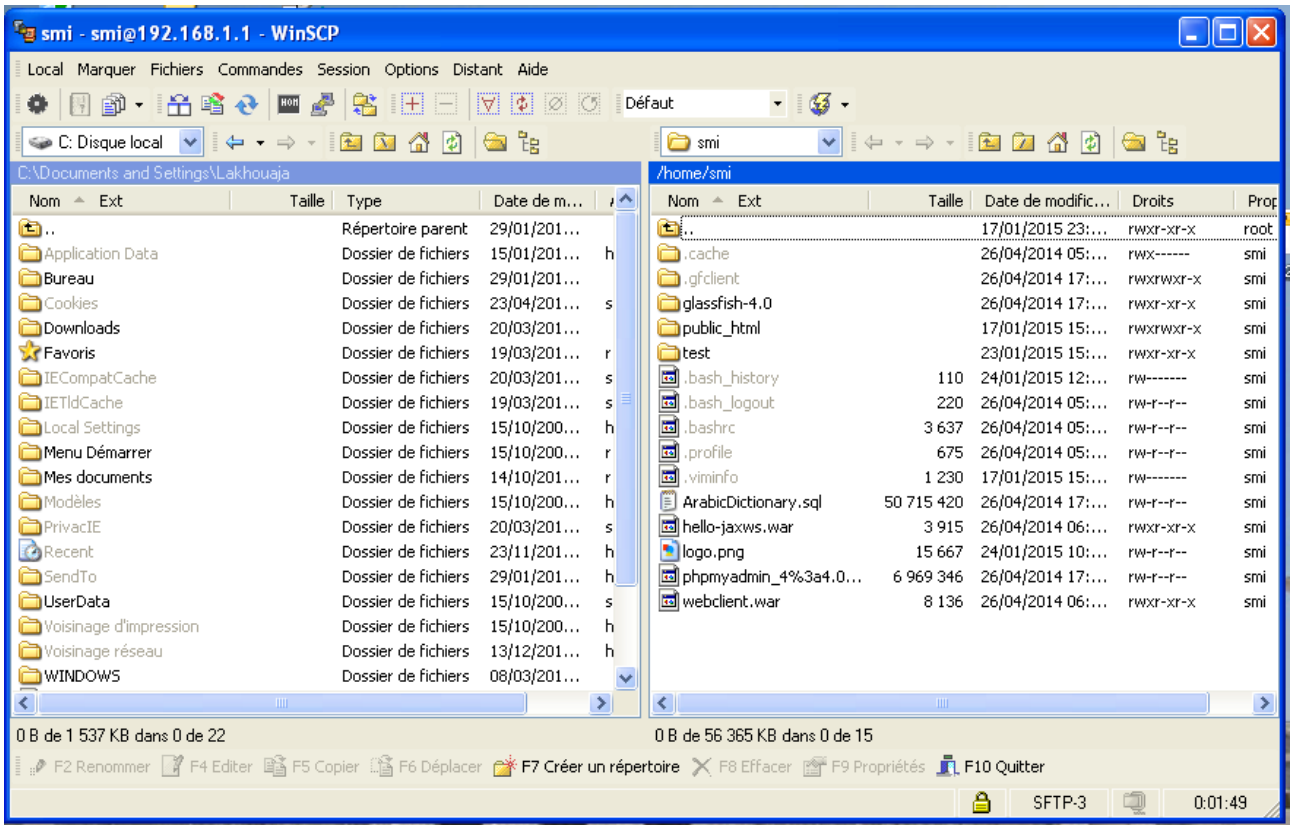
Remarque

Si votre navigateur supporte le protocole **sftp**, vous pouvez tapez par exemple : `sftp://smi@192.168.56.2/`

5.3.3 Connexion à partir d'un client Windows

Sous Windows il existe l'application **winscp** disponible en téléchargement à partir su site officiel `http://winscp.net`. Son interface graphique se présente comme suit :





5.3.4 Copie vers le serveur

Pour copier un fichier ou un répertoire dans le serveur ssh, vous pouvez utiliser la commande **scp** (analogue à la commande **cp** de Linux). Son utilisation est comme suit :

```
scp fichier1 fichier2 ... smi@192.168.56.2:
```

Pour copier un répertoire, il faut simplement ajouter l'option `-r` :

```
scp -r Rep smi@192.168.56.2:
```

Remarque : il ne faut pas oublier **:**, sinon la copie se fera en local (utilisation de **cp**).

Chapitre 6

Partage de dossiers et d'imprimantes

6.1 Introduction

Le partage de dossiers et d'imprimantes permet de :

- réduire le coût d'investissement ;
- mutualiser les ressources.

Pour partager des dossiers entre des machines Unix/Linux on utilise le protocole NFS (Network File System).

Pour partager des dossiers et des imprimantes entre des machines Unix/Linux et des machines Windows on utilise le service samba.

6.2 Le protocole NFS

C'est un protocole qui fonctionne suivant le modèle client/serveur :

- un serveur met des dossiers à la disposition des machines sur le réseau suivant des droits d'accès.
- d'autres machines peuvent monter ces dossiers. Qui seront vus comme des dossiers locaux.

6.3 Côté serveur

Il faut installer e serveur NFS, en tapant la commande :

```
# sudo apt-get install nfs-kernel-server
```

6.3.1 Configuration

Le fichier de configuration est /etc/exports. On indique dans ce fichier la liste des répertoires à exporter (partagés) et les noms des machines autorisées à les utiliser.

Exemple :

```
/home machine1(rw,sync,no_subtree_check)    *(ro,sync,no_subtree_check)
/projet machine1(rw,sync,no_root_squash,no_subtree_check)
/test *(ro,sync,no_subtree_check)
```

Explication :

- **machine1** peut monter /**home** en lecture/écriture (rw) ;
- toutes les autres machines du réseau peuvent monter /**home** en lecture seulement (ro) ;
- **machine1** peut monter /**projet** en lecture/écriture (rw) ;
- toutes les machines du réseau peuvent monter /**test** en lecture seule (ro).

On peut utiliser des noms ou adresses IP pour les machines.

Une fois le fichier /**etc**/**exports** bien configurer il faut redémarrer (ou relancer) **nfs** :

```
# sudo service nfs-kernel-server restart
```

ou bien

```
# sudo service nfs-kernel-server reload
```

Principales options du fichier exports [1] :

Option	Signification
ro	read-only (accès en lecture seule au répertoire exporté)
rw	read-write : le client accède au répertoire en lecture/écriture
root_squash	convertit les UID/GID root, en utilisateur anonyme. L'administrateur de la machine cliente ne peut pas modifier le contenu des répertoires et des fichiers.
no_root_squash	désactive la conversion des UID/GID root.
all_squash	convertit tous les UID/GID en utilisateurs anonymes. Utile pour exporter avec NFS des répertoires publics.
sync	ne répondre aux requêtes qu'après l'exécution de tous les changements sur le support réel.
no_subtree_check	annule la vérification des sous-répertoires

Remarque : les options doivent être séparées par des virgules, SANS ESPACE.

Pour plus d'options, veuillez consulter le manuel du fichier export (`man exports`).

6.4 Côté client

Pour pouvoir monter des répertoires, il faut installer la package **nfs-common** :

```
#sudo apt-get install nfs-common
```

En tant qu'administrateur du système, pour monter un répertoire distant, il faut utiliser la commande **mount** avec l'option **-t nfs**.

Exemple :

```
#mount -t nfs nom_machine:/home /test
```

ou

```
#mount -t nfs 192.168.56.2:/home /test
```

1. manuel de **exports** : `man exports`

6.4.1 Montage au démarrage

Pour monter un répertoire au démarrage du système, il suffit d'ajouter les renseignements nécessaire au fichier **/etc/fstab**

Par exemple :

```
nom_machine:/home /home nfs auto,rw,user 0 0
```

6.5 NFS et la Sécurité

NFS n'est pas un protocole très sécurisé :

- l'authentification des clients repose uniquement sur le nom de domaine ou l'adresse IP ;
- l'identification des utilisateurs repose sur le « user id » sur le poste client => usurpation possible ;
- le transfert des données est non crypté ;
- utilisation recommandée en intranet isolé, protégé de l'internet par un Firewall.

6.6 Le protocole SAMBA

Samba permet de partager des répertoires et des imprimantes entre Linux et d'autres systèmes Windows et Mac OS.

6.6.1 Installation

En ligne de commandes, il suffit de taper la commande :

```
#sudo apt-get install samba
```

6.6.2 Configuration

Le fichier principal de configuration de samba est : **/etc/samba/smb.conf**.

Avant de modifier le fichier de configuration, il faut le sauvegarder par prudence (`cp smb.conf smb.conf.old`).

Ce fichier est organisé en sections. L'administrateur **root** peut éditer, modifier et ajouter des sections, pour définir de nouvelles ressources à partager.

Une section commence par un mot entre crochets et se termine lorsqu'une autre section commence.

Exemple de sections :
[global]
#ensemble de directives

[homes]
#ensemble de directives

Remarque :

D'une façon générale, les permissions de partage définies dans les sections ne peuvent pas outrepasser les permissions des fichiers du serveur hôte.

Pour plus d'informations concernant le fichier smb.conf, veuillez consulter le manuel en ligne :

```
man smb.conf
```

Vérification des changements

L'outil **testparm**, permet de tester la syntaxe du fichier de configuration et de détecter les erreurs. Il est recommandé de le lancer systématiquement lors de la modification de **smb.conf**.

Activation des changements

A chaque changement effectué dans **smb.conf**, il faut relancer les démons **smbd** et **nmbd**. Commande :

```
#service smbd restart
```

suivie de :

```
#service nmbd restart
```

6.6.3 Les principaux paramètres de smb.conf

Dans le tableau suivant, nous donnons les principaux paramètres de **smb.conf** :

paramètre	**valeur par défaut**	**Description**
path =		chemin du répertoire à partager
comment =		texte visible dans le voisinage réseau client
guest ok = yes|no	no	permettre l'accès sans authentification
valid users =	tous	liste des utilisateurs autorisés à se connecter à la ressource
printable = true|false	false	partage d'un service d'impression et non d'un répertoire.
writeable = yes|no	no	permet ou non l'écriture sur le répertoire, contraire de read only
browseable =	yes	visibilité du partage par tous, même les utilisateurs non autorisés
create mask =	0744	droits maxi accordés à un fichier créé dans la ressource ces droits seront en intersection (and) avec les droits Linux (umask)
directory mask =	0755	droits maxi accordés à un répertoire créé dans la ressource ces droits seront en intersection (and) avec les droits Linux (umask)

Commande umask

L'umask permet d'attribuer des permissions aux fichiers et répertoires créés par l'utilisateur. Il se présente sous la forme de 4 chiffres. La valeur par défaut de l'umask est 0022. Pour obtenir les permissions qui seront utilisées, il faut appliquer la règle suivante :

- pour les fichiers, il faut soustraire le umask de 666.
  Par exemple $666 - 0022 = 644$ ce qui donne les droits `rw-r--r--`
- pour les répertoires, il faut soustraire le umask de 777.
  Par exemple $777 - 0022 = 755$ ce qui donne les droits `rwxr-xr-x`

Si l'utilisateur veut que les nouveaux fichiers soient créés avec les droits `rw-------` et que les nouveaux répertoires soient créés avec les droits `rwx------`, il doit utilisé le masque **0077**. Pour cela, il doit taper la commande :

```
umask 0077
```

ou tout simplement :

```
umask 77
```

Remarque

umask accepte les symboles (r, w et x) comme **chmod**.

La commande « `umask 77` » peut être utilisée comme suit :

```
umask u=rwx,g=,o=
```

6.6.4 La section globale

```
[global]
# donner le meme nom de groupe de travail que celui des stations Windows
(Voisinage reseau/identification)
workgroup = SMI
# restreindre par sécurité les sous-réseaux autorisés à se connecter au serveur
# ici on se limite aux adresses réseau privé 192.168.1.0 et à l'interface "loopback"
hosts allow = 192.168.1.      127.
# on peut exclure des machines de l'accès au réseau
hosts allow = 192.168.1.  EXCEPT 192.168.1.125
# d'autres possibilités existent : voir le manuel man smb
```

6.6.5 Le répertoire personnel

```
[homes]
#accès au répertoire personnel de chaque utilisateur.
#la valeur du champ "comment" apparaîtra dans le voisinage réseau
#inutile pour cette section de préciser le path, c'est celui de l'utilisateur, en fait /home/%u

comment = Répertoire personnel
```

```
browsable = no
writable = yes
create mode = 0700
```

6.6.6 Rendre un répertoire public

Pour rendre un répertoire accessible par tous le monde, il faut tout d'abord le créer ou vérifier qu'il existe.

`# mkdir /home/partage`

`ls -ld /home/partage`

doit renvoyer les droits par défaut `drwxr-xr-x`, sinon il faut les changer en tapant la commande :

`chmod 755 /home/partage`

ou son équivalent

`chmod u=rwx,go=rx /home/partage`

pour y ajouter les permissions d'accès pour tous. Ensuite, il faut ajouter une nouvelle section « **[partage]** » comme suit :

```
[partage]
path = /home/partage
browsable = yes
writable = no
guest ok = yes
```

Si on veut rendre ce répertoire partagé en écriture aussi, il faut modifier les droits d'accès du répertoire et modifier la section [partage]

`chmod 777 /home/partage`

```
[partage]
path = /home/partage
browsable = yes
writable = yes
guest ok = yes
create mode = 0755
```

6.6.7 Utilitaires SAMBA

testparm : permet la validation du fichier de configuration de Samba.

smbclient : client Linux/Unix similaire à FTP permettant de se connecter à des partages Samba.

smbpasswd : permet à un administrateur de modifier les mots de passe chiffrés utilisés par Samba.

smbstatus : dresse l'état des connexions aux partages d'un serveur Samba.

6.6.8 Ajout d'un utilisateur samba

Pour permettre à un utilisateur de se connecter à son répertoire personnel à partir d'autres machines, il faut l'ajouter en tant qu'utilisateur samba en tapant la commande suivante :

smbpasswd -a utilisateur

Par exemple :

smbpasswd -a smi

6.6.9 Problème de connexion avec Windwos

Sous Windows, si on se connecte avec un utilisateur 1 (par exemple **smi**) et on veut se connecter avec un autre utilisateur (par exemple **sma**), la connexion ne réussisse pas. Pour cela, il faut supprimer la connexion à l'utilisateur 1, en tapant la commande (sous un invite de commande) :

```
net use \\nom-partage\utilisateur /delete
```

Par exemple :

```
net use \\Ubuntu\smi /delete
```

Ou **Ubuntu** est le nom de partage et **smi** est le nom de l'utilisateur déjà connecté.

6.6.10 Connexion à partir d'un client Linux

Pour se connecter à partir d'un client Linux en utilisant la commande **smbclient**, il faut taper la commande

smbclient //nom-machine/repertoire

Par exemple :

smbclient //192.168.56.2/partage

Pour se connecter en utilisant le compte d'un utilisateur qui s'appelle **smi**, il faut taper la commande :

smbclient //192.168.56.2/smi -U smi

ensuite, on saisit le mot de passe.

On peut utiliser la commande **mount** pour monter un répertoire partagé (fonctionne sous root) :

mount -o username=smi //192.168.56.2/smi Rep

Ou **Rep** est le répertoire de montage.

Chapitre 7

Domain Name Service (DNS)
Service de Nom de domaines

7.1 Introduction

Domain Name Service (DNS) est un service qui relie les adresses IP et les noms de domaines entre eux. Sous Linux, le DNS est géré par **BIND** (Berkeley Internet Name Domain) « paquet **bind9** ».

Pour les petits réseaux, il suffit d'utiliser le fichier **/etc/hosts** que vous avez vu en semestre 5.

Exemple d'un fichier /etc/hosts :

```
127.0.0.1       localhost
127.0.1.1       alkhalil.ump.ma     alkhalil
192.168.56.2    smi.ump.ma          smi
192.168.56.2    www.smi.ump.ma
192.168.56.2    sma.ump.ma          sma
192.168.56.2    www.sma.ump.ma
```

7.2 Installation

Dans un terminal, tapez la commande :

```
sudo apt-get install bind9
```

7.3 Configuration

Bind peut être configuré de plusieurs façons. Il peut être configuré pour être un serveur :

de cache : dans ce cas, il sert pour stocker les informations concernant les requêtes sur les noms de domaines ;

principale : il lit les données pour une zone à partir d'un fichier stocké localement et il est autoritaire pour cette zone ;

secondaire : il obtient les données concernant une zone à partir d'un serveur de noms autoritaire pour cette zone.

Remarque : un serveur de nom peut être autoritaire pour une zone et secondaire pour une autre zone.

Le fichier de configuration principal de **bind** est **/etc/bind/named.conf**.

7.3.1 Commentaires

Des commentaires peuvent être utilisés :
- /* commentaire de type C (peut occuper plusieurs lignes) */
- // commentaire de type java, C++
- # commentaire de type shell

7.4 Configuration comme serveur principale

Dans cette section, nous allons configurer **bind** pour être un serveur principale du domaine **smi6.net**.

Le fichier **/etc/bind/named.conf** contient la ligne :

```
include "/etc/bind/named.conf.local";
```

Cette ligne veut dire, que pour ajouter une zone locale, il faut l'ajouter dans le fichier **/etc/bind/named.conf.local**.

Pour ajouter la zone **smi6.net**, il faut ajouter dans le fichier **/etc/bind/named.conf.local** les lignes suivantes :

```
zone "smi6.net" {
        type master;
        file "/etc/bind/db.smi6.net";
};
```

Remarque : le plus simple pour créer le fichier **/etc/bind/db.smi6.net**, est d'utiliser un fichier qui existe déjà. Par exemple :

```
sudo cp /etc/bind/db.local /etc/bind/db.smi6.net
```

Dans notre cas le fichier **/etc/bind/db.smi6.net** contiendra les instructions suivantes :

```
$TTL    86400
@       IN    SOA     smi6.net.     dns.smi6.net. (
        250120152     ; Serial
        3600             ; Refresh (1 heure)
        86400         ; Retry (1 jour)
        2419200       ; Expire (28 jours)
        86400 )       ; Minimum (1 jour)

        IN  A    192.168.1.1
;
@    IN  NS   dns.smi6.net.
dns  IN  A    192.168.1.1
r1   IN  A    192.168.1.1
pc1  IN  A    192.168.1.2
```

Signification des différents champs

- **TTL** (Time To Live) : détermine le temps, en secondes, durant lequel les informations seront conservées dans le cache.
- **SOA** (Start Of Authority) : indique le début d'un enregistrement.
- **NS** (Name Server) : identifie un serveur de nom pour un domaine.
- **A** (internet Address) : adresse internet.
- **@** : désigne le nom du domaine actuel. Il ne faut pas oublier le point (.) après le nom de domaine.
- **;** : commentaire.

Signification des valeurs numériques

Serial : numéro de série. Un numéro unique qui identifie la version du fichier de la zone. En général, vous avez cette valeur sous la forme de date de modification du fichier suivie d'un numéro (250120152 : 25/01/2015+2). Dans la plus part des cas, vous trouverez la date sous la forme AAAAMMJJ (250120152 devient : 201501252).

Refresh : (rafraîchir) le temps en secondes que doit mettre un serveur DNS secondaire pour vérifier le numéro de série.

Retry : le temps en secondes que doit attendre un serveur DNS secondaire après une mauvaise requête de rafraîchissement.

Expire : le temps en secondes qu'un serveur DNS secondaire doit utiliser les données avant de faire un rafraîchissement. Cette valeur doit être grande.

Minimum : le temps en secondes qui doit être utilisé pour le TTL.

Redémarrage de bind

Une fois les changements effectués, redémarrez le service **bind** en tapant la commande :

```
sudo service bind9 restart
```

7.5 Côté client

Pour configurer le client, il faut éditer le fichier de configuration de la résolution de noms **/etc/resolv.conf**.

Exemple :

domain smi6.net
nameserver 192.168.1.1

7.5.1 Vérification

A partir d'un client, il suffit d'utiliser la commande **ping** pour vérifier la connexion aux différentes machines. Par exemple :

`ping r1.smi6.net`

7.5.2 nslookup

La commande **nslookup** permet l'interrogation d'un serveur DNS.

Exemple d'utilisation :

La commande :

`nslookup pc1.smi6.net`

fournira le résultat :

```
Server: 192.168.1.1
Address: 192.168.1.1#53

Name: pc1.smi6.net
Address: 192.168.1.2
```

53 correspond au numéro de port utilisé par le serveur DNS (voir fichier **/etc/services**).

7.5.3 Fichier de la zone inverse (Reverse Zone)

Il permet au serveur DNS de faire la résolution d'adresses vers des noms.

Il faut ajouter dans le fichier **/etc/bind/named.conf.local** les lignes suivantes :

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
```

Ensuite, il faut créer le fichier **/etc/bind/db.192** :

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

De la même façon que pour le fichier **/etc/bind/db.smi6.net**, il faut éditer le fichier **/etc/-bind/db.192** pour devenir comme suit :

```
;
; Fichier BIND inverse pour le reseau local 192.168.1.0
;
$TTL    604800
@   IN  SOA smi6.net. dns.smi6.net. (
            090320152      ; Serial
               604800      ; Refresh
                86400      ; Retry
              2419200      ; Expire
               604800 )    ;
;
@   IN  NS  dns.
1   IN  PTR dns.smi6.net.
2   IN  PTR pc1.smi6.net.
```

Une fois le fichier inverse créé, redémarrez le service **bind** :

```
sudo service bind9 restart
```

7.6 Configuration d'un serveur secondaire

Une fois le serveur primaire configuré, il faut avoir un autre serveur secondaire pour que la zone soit toujours disponible même si le serveur principale tombe en panne.

Nous allons supposer que nous disposons d'un autre serveur avec l'adresse **192.168.1.10**.

Dans le serveur principale il faut activer le transfert en ajoutant les options **allow-transfer** et **also-notify** au fichier **/etc/bind/named.conf.local** comme suit :

```
zone "smi6.net" {
    type master;
    file "/etc/bind/db.smi6.net";
    allow-transfer { 192.168.1.10; };
    also-notify { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.1.10; };
    also-notify { 192.168.1.10; }
};
```

Dans le serveur principal, redémarrez le service **bind** :

```
sudo service bind9 restart
```

Dans le serveur secondaire, il faut installer **bind9** de la même façon que pour le serveur principale. Puis il faut ajouter les déclarations suivantes dans le fichier **/etc/bind/named.conf.local** :

```
zone "smi6.net" {
    type slave;
    file "/etc/bind/db.smi6.net";
    masters { 192.168.1.1; };
};

zone "1.168.192.in-addr.arpa" {
    type slave;
    file "db.192";
    masters { 192.168.1.1; };
};
```

Dans le serveur secondaire, redémarrez le service **bind** :

```
sudo service bind9 restart
```

Chapitre 8

Sécurité

8.1 Introduction

La sécurité doit être prise en considération lors de l'installation et l'utilisation d'un ordinateur.

Les attaques touchent généralement les trois composantes suivantes :
- La couche d'application
- Le système d'exploitation
- La couche réseau

Cependant on distingue différentes attaques au sein d'un réseau dû à la faiblesse des composants :

- faiblesses d'authentification ;
- mauvaises configurations.
- faiblesses d'implémentation ou de bogues ;
- faiblesses liées aux protocoles.

8.2 Authentification

La gestion des utilisateurs est fondamentale dans la sécurité d'un système informatique. De mauvaises privilèges ou un mauvais mot de passe peuvent compromettre la sécurité d'un ordinateur.

8.2.1 Profile des utilisateurs

Lors de la création d'un nouveau utilisateur avec la commande **adduser** (par exemple **adduser smi**), le répertoire personnel de l'utilisateur **smi** est créé avec les droits `drwxr-xr-x`. Il faut enlever les droits de lecture pour les autres :

```
sudo chmod 750 /home/smi
```

Pour mettre cette valeur par défaut lors de la création d'un nouveau utilisateur avec la commande **adduser**, il faut modifier la valeur de la variable `DIR_MODE` dans le fichier **/etc/adduser.conf** de la façon suivante :

```
DIR_MODE=0750
```

8.2.2 Mots de passe

Pour éviter les attaques qui utilisent un dictionnaire, le mot de passe doit être fort. Il doit :

— comporter des lettres minuscules et majuscules, des nombres et d'autres caractères ;
— comporter au moins 8 caractères ;

Il ne doit pas comporter :

— le nom ou le prénom de l'utilisateur ;
— la date de naissance de l'utilisateur ;
— un mot du dictionnaire.

8.3 Les Firewall (Pare Feu)

Un pare-feu est un ensemble matériel ou logiciel qui trie les paquets qui circulent par son intermédiaire en provenance ou vers le réseau local, et ne laisse passer que ceux qui vérifient certaines conditions.

C'est un système de protection dédié à la sécurité d'un réseau.

Les noyaux Linux contiennent le système **Netfilter** pour manipuler le trafic réseau. Pour accepter, manipuler ou rejeter un paquet, on utilise **iptables**.

8.4 iptables

iptables est très utilisé pour mettre en place un pare-feu. Elle utilise 4 ou 5 tables (le nombre dépend du système). Une table permet de définir un comportement précis de **Netfilter**. En fait, c'est un ensemble de chaînes, elles-mêmes composées de règles.

Les tables sont :

1. **Filter** : c'est la table par défaut. Elle s'utilise sans l'option **-t** et contient les chaînes :
   — **INPUT :** pour les paquets destinés aux sockets local ;
   — **FORWARD :** pour les paquets routés ;
   — **OUTPUT :** pour les paquets générés localement.
2. **NAT** : est consultée quand un paquet qui crée une nouvelle connexion est rencontré. Elle consiste en trois chaînes :
   — **PREROUTING :** pour les paquets qui entrent ;
   — **OUTPUT :** pour les paquets générés localement avant le routage ;
   — **POSTROUTING :** pour les paquets qui sortent.
3. **Mangle** : sert a modifier d'autres paramètres des paquets IP (notamment le champ ToS — Type Of Service — et les options). Elle consiste en cinq chaînes :
   — **PREROUTING :** paquets entrant avant le routage ;
   — **OUTPUT :** pour les paquets générés localement avant le routage ;
   — **INPUT :** paquets arrivant au système lui même ;
   — **FORWARD :** paquets routés via le système ;
   — **POSTROUTING :** pour les paquets qui sortent
4. **Raw** : contient les chaînes :

— **PREROUTING :** pour les paquets arrivant de n'importe quelle interface réseau
— **OUTPUT :** pour les paquets générés par les processus locales

5. **security** (dans les machines virtuelles **netkit**, cette table n'est pas disponible).

Dans ce qui suit nous allons utiliser la table par défaut.

8.4.1 Initialisation des tables

On vide les chaînes au niveau de la table **Filter** :

```
pc1: # iptables -F
```

On supprime les éventuelles chaînes personnelles :

```
pc1: # iptables -X
```

8.4.2 Blocage des tables

Maintenant faisons pointer par défaut les chaînes de la table **Filter** sur **DROP** (Rejet) :

```
pc1: # iptables -P INPUT DROP
pc1: # iptables -P OUTPUT DROP
pc1: # iptables -P FORWARD DROP
```

Les entrées et les sorties sont bloquées.

8.4.3 Test de sortie

Un **ping** de **pc1** vers **pc2** donne :

```
pc1: # ping -c 1 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
ping: sendmsg: Operation not permitted

--- 192.168.100.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Les paquets ne sortent pas de **pc1**.

8.4.4 Test d'entrée

Un **ping** de **pc2** vers **pc1** donne :

```
pc2: # ping -c 1 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Les paquets arrivent sur **pc1** et sont rejetés. Il suffit de le vérifier avec **tcpdump**.

8.4.5 Test vers la boucle locale

Même un **ping** vers **localhost** est rejeté :

```
pc1: # ping -c 1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted


--- localhost ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

8.4.6 Examen de la table Filter

Examinons l'état de la table **Filter** :

```
pc1: # 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
```

8.4.7 Autorisation de la boucle locale

On autorise des entrées locales :

```
pc1: # iptables -A INPUT -i lo -j ACCEPT
```

On autorise des sorties locales :

```
pc1: # iptables -A OUTPUT -o lo -j ACCEPT
```

Alors si on fait un **ping** sur la machine elle-même on voit que sa marche :

```
pc2:~# ping -c 1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.093 ms

--- localhost ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.093/0.093/0.093/0.000 ms
```

Examinons l'état de la table **Filter** :

```
#l'option -v veut dire verbose (bavard), donne plus de détails
pc1: # iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   14  1176 ACCEPT     all  --  lo     any     anywhere             anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 4 packets, 336 bytes)
 pkts bytes target     prot opt in     out     source               destination
   14  1176 ACCEPT     all  --  any    lo      anywhere             anywhere
```

8.4.8 Autoriser le trafic d'une connexion déjà établie

Pour autoriser une connexion déjà ouverte d'envoyer et de recevoir du trafic :

```
pc1: # iptables -A INPUT -m conntrack -ctstate ESTABLISHED -j ACCEPT
```

```
pc1: # iptables -A OUTPUT -m conntrack -ctstate ESTABLISHED -j ACCEPT
```

8.4.9 Ouverture de quelques ports/services

Pour autoriser les connexion au serveur **SSH**, il faut :

- autoriser les entrées des requêtes au port **ssh**
  `pc1: # iptables -A INPUT -p tcp -dport ssh -i eth0 -j ACCEPT`
- autoriser les sorties des requêtes utilisant **ssh**
  `pc1: # iptables -A OUTPUT -p tcp -dport ssh -o eth0 -j ACCEPT`

Pour autoriser l'envoi et la réception de messages **ICMP**, il faut :

- autoriser les entrées des requêtes utilisant le protocole **icmp**
  `pc1: # iptables -A INPUT -p icmp -i eth0 -j ACCEPT`
- autoriser les sorties des requêtes utilisant le protocole **icmp**
  `pc1: # iptables -A OUTPUT -p icmp -o eth0 -j ACCEPT`

Chapitre 9

Commandes CISCO de base

9.1 Configuration du nom d'hôte IOS

En mode d'exécution privilégié (commande **enable**), accédez au mode de configuration globale en entrant la commande configure terminal :

```
Router>enable
Router#configure terminal
```

Après exécution de cette commande, l'invite devient :

```
Router(config)#
```

En mode de configuration globale, entrez le nom d'hôte :

```
Router(config)#hostname SMIRouter
```

Après exécution de cette commande, l'invite devient :

```
SMIRouter(config)#
```

Par exemple, pour supprimer le nom attribué à un périphérique, utilisez :

```
SMIRouter(config)# no hostname
Router(config)#
```

La commande **no hostname** rétablit sur le routeur le nom d'hôte par défaut « **Router** ».

9.2 Mots de Passe

Les mots de passe présentés ici sont les suivants :

- **Mot de passe de console :** limite l'accès au périphérique par une connexion console.
- **Mot de passe enable** - limite l'accès au mode d'exécution privilégié.
- **Mot de passe « enable secret » :** chiffré, limite l'accès au mode d'exécution privilégié

Il est recommandé d'utiliser des mots de passe différents pour chacun de ces niveaux d'accès.

9.2.1 Mot de passe de console

Vous utilisez les commandes suivantes en mode de configuration globale pour définir un mot de passe pour la ligne de console :

```
Router(config)#line console 0
Router(config-line)#password mot_de_passe
Router(config-line)#login
```

A partir du mode de configuration globale :

- la commande **line console 0** permet d'entrer en mode de configuration de ligne pour la console. Le zéro sert à représenter la première (et le plus souvent l'unique) interface de console d'un routeur.
- La deuxième commande, password mot_de_passe, spécifie un mot de passe sur une ligne.
- Enfin, la commande **login** configure le routeur pour exiger une authentification à l'ouverture de session. Après activation de **login** et définition d'un mot de passe, le périphérique invitera l'utilisateur à entrer un mot de passe.

Le message demandant le mot de passe apparaîtra désormais chaque fois qu'un utilisateur tentera d'accéder au port de console.

9.2.2 Mots de passe enable et enable secret

Pour augmenter la sécurité, utilisez la commande la commande **enable secret**.

La commande suivante est utilisée pour définir un mot de passe :

```
Router(config)#enable secret  mot_de_passe
```

9.2.3 Chiffrement de l'affichage des mots de passe

Une autre commande utile permet d'empêcher l'affichage des mots de passe en clair lorsqu'un utilisateur consulte les fichiers de configuration. Il s'agit de la commande :

```
service password-encryption
```

Cette commande provoque le chiffrement des mots de passe déjà configurés. La commande **service password-encryption** applique un chiffrement simple à tous les mots de passe non chiffrés. Ce chiffrement ne s'applique pas aux mots de passe transmis sur le support pendant la configuration. Le but de cette commande est d'empêcher les personnes non autorisées de lire les mots de passe dans le fichier de configuration.

Si vous exécutez la commande **show running-config** ou **show startup-config** avant la commande **service password-encryption**, les mots de passe non chiffrés sont visibles dans les informations fournies par le périphérique sur sa configuration. Dès que vous exécutez la commande **service password-encryption**, IOS applique le chiffrement aux mots de passe. Par la suite, les mots de passe déjà chiffrés le restent même si vous supprimez le service password-encryption (en annulant la commande).

9.3 Configuration des interfaces Ethernet d'un routeur

Chaque interface Ethernet doit avoir une adresse IP et un masque de sous-réseau pour router les paquets IP.

Pour configurer une interface Ethernet, procédez comme suit :

1. Passez en mode de configuration globale.
2. Passez en mode de configuration d'interface.
3. Spécifiez l'adresse et le masque de sous-réseau de l'interface.
4. Activez l'interface.

```
Router(config)#interface FastEthernet 0/0
Router(config-if)#ip address adresse_ip masque_réseau
Router(config-if)#no shutdown
```

9.3.1 Activation de l'interface

Par défaut, les interfaces sont désactivées. Pour activer une interface, entrez la commande **no shutdown** à partir du mode de configuration d'interface. Si vous devez désactiver une interface pour des opérations de maintenance ou de dépannage, utilisez la commande **shutdown**.

9.4 Routage statique

9.4.1 Commande ip route

La commande de configuration d'une route statique est **ip route**. La syntaxe simplifiée pour configurer une route statique est :

```
Router(config)#ip route adresse_réseau masque {adresse-ip | interface-sortie }
```

Les paramètres suivants sont utilisés :

- **adresse_réseau :** adresse réseau de destination du réseau distant à ajouter à la table de routage.
- **masque :** masque de sous-réseau du réseau distant à ajouter à la table de routage.

Un des paramètres suivants ou les deux doivent également être utilisés :

- **adresse-ip** : communément considérée comme l'adresse IP du routeur du tronçon suivant.
- **interface-sortie** : interface sortante à utiliser pour le transfert de paquets vers le réseau de destination.

Exemple :

```
Router#debug ip routing
Router#conf t
Router(config)#ip  route  172.16.1.0  255.255.255.0   172.16.2.2
```

Pour afficher la table de routage utilisez la commande : **show ip route**.

9.5 Routage dynamique (RIPv1)

9.5.1 Activation du protocole RIP

Pour passer en mode de configuration du routeur pour le protocole **RIP**, entrez la commande **router rip** à l'invite de configuration globale.

```
Router(config-router)#
```

Pour activer le routage RIP pour un réseau, utilisez la commande **network** dans le mode de configuration du routeur et entrez l'adresse réseau par classe de chaque réseau directement connecté.

```
Router(config-router)#network   réseau-connecté-à-l'interface
```

La commande **network** :

- active le protocole RIP sur toutes les interface qui appartiennent à un réseau spécifique. Les interfaces associées envoient et reçoivent maintenant les mises à jour RIP.
- Annonce le réseau spécifié dans les mises à jour de routage RIP envoyées aux autres routeurs toutes les 30 secondes.

Remarque : si vous entrez une adresse de sous-réseau, l'IOS la convertit automatiquement en adresse réseau par classe. Par exemple, si vous entrez la commande **network 192.168.1.32**, le routeur la convertira en **network 192.168.1.0**.

Exemple :

```
Router(config)#router rip
Router(config-router)#network 192.168.4.0
Router(config-router)#network 192.168.5.1
```

Dans cet exemple, vous avez entré une adresse IP d'interface au lieu de l'adresse réseau par classe. Notez que l'IOS n'affiche pas de message d'erreur, mais corrige l'entrée et indique l'adresse réseau par classe. La vérification ci-dessous le prouve.

```
Router#show running-config
```

9.5.2 Table de routage

La commande **show ip route** permet d'afficher la table de routage.

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.2.254, 00:00:27, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/1
```