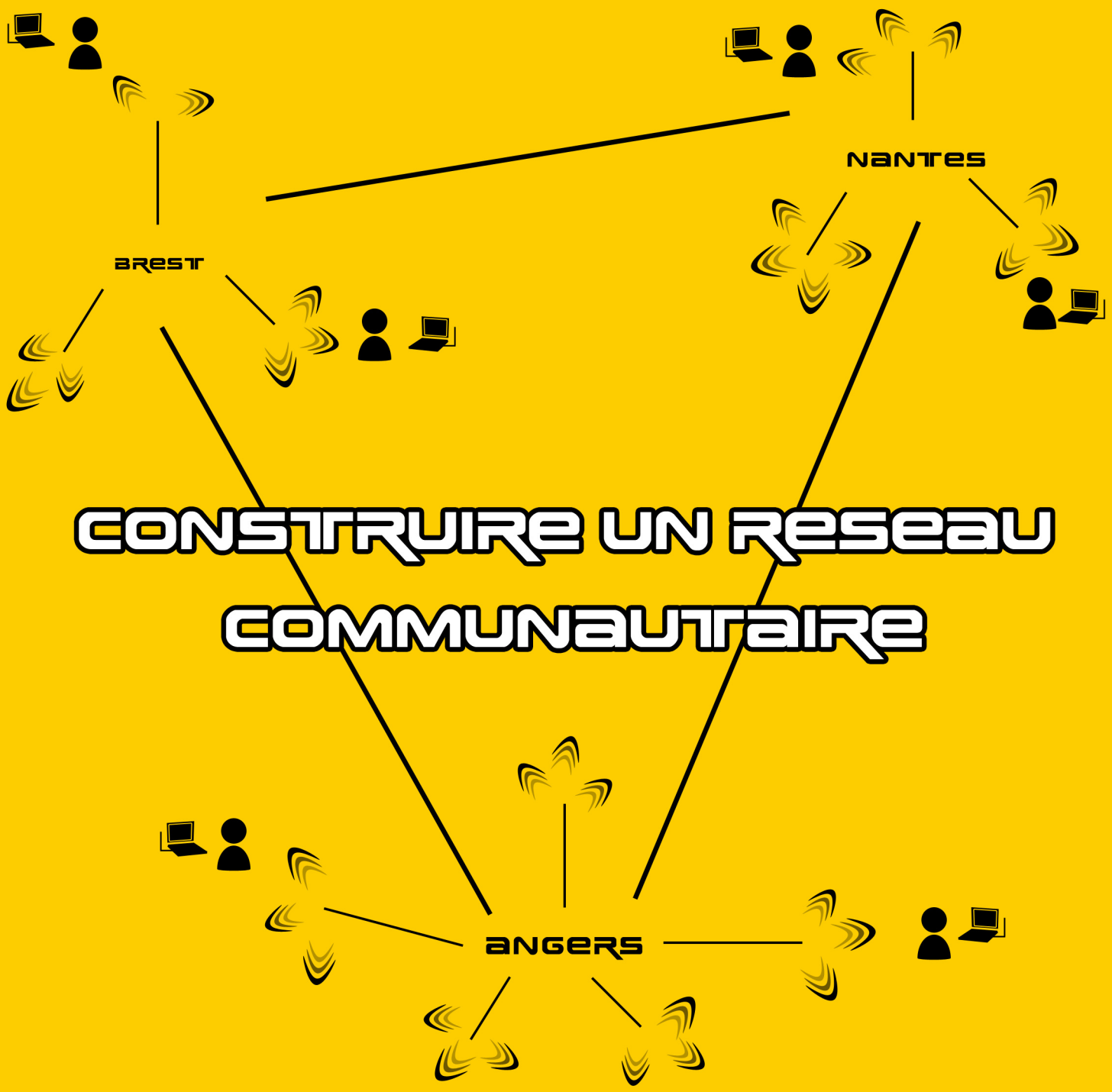



# CONSTRUIRE UN RESEAU COMMUNAUTAIRE

# Sommaire

# A propos de cette documentation

Bienvenue dans la documentation libre "Construire un réseau communautaire".
Cette documentation a pour objectif de vous aider à installer, configurer et administrer un réseau communautaire de type WLAN (sans-fil).
Vous y apprendrez comment: configurer le réseau sur une distribution Debian GNU Linux, installer un serveur Web, installer et configurer des liens VPNs, installer et configurer un serveur DNS, mettre en place différents services réseau Windows grâce à Samba, sécuriser votre réseau grâce à Netfilter (iptables) et FreeRadius, ...
A la fin de cette documentation vous serez en mesure de mettre en place un réseau étendu composé d'une multitude de sous-réseaux reliés entre eux via des liens VPN, réseau sécurisé et permettant notamment l'authentification des utilisateurs grâce à un serveur Radius. Ce réseau global pourra accepter des clients mobiles qui pourront se connecter depuis n'importe quel point du réseau.

Toute cette documentation sera orientée autour de la mise en place du réseau pour l'association Angers-Wireless, les exemples et fichiers de configuration sont ceux du serveur et des noeuds de l'association.

## Public concerné

Cette documentation s'adresse principalement à des informaticiens ou à des personnes ayant déjà des connaissances dans le domaine des réseaux, en effet, dans cette documentation les concepts "de base" des réseaux ne seront pas ou peu présents, Les étudiants pourront y trouver différentes informations leurs permettant de mener à terme travaux pratiques et projets.

## Conditions préalables

Pour tirer pleinement parti de cette documentation:

- Vous devez avoir des connaissances minimales dans le domaine des systèmes d'exploitation Linux / UNIX.
- Vous devez déjà maîtriser les principes fondamentaux des technologies réseau actuelles.

## Caractéristiques de cette documentation

Cette documentation à été réalisée par François GERTHOFFERT et Thomas BAUGE étudiants en formation CPI Informatique et Réseaux.

## Conventions de notation

Les commandes à taper dans une console sont présentées **en gras** dans un cadre grisé et précédées d'un numéro de ligne. Un détail des commandes est présent en dessous du cadre gris.

Les résultats de commandes sont présentés en italique dans un cadre de fond gris.

```
1:# uptime
17:35:10 up 5 days, 20:47,  2 users,  load average: 0.05, 0.03, 0.00
```

Ligne 1: Durée depuis le dernier reboot

Les fichiers de configuration sont présentés *en italique* dans un cadre bleu, les commentaires présents dans les fichiers de configuration sont généralement précédés du caractère "#".

Tous les mots de passe des fichiers de configuration seront remplacés par des **xxxxxx** ; à vous de choisir vos propres mots de passe.

*Fichier /etc/raddb/clients*

```
# Client Name         Key
#----------------     ----------
localhost            xxxxxx
```

## Aperçu des chapitres et des annexes

Cette documentation est constituée d'explications théoriques et d'exemples de configuration dont le but est de vous apprendre à mettre en place un réseau de type WLAN.
La documentation est organisée comme suit:

- Le chapitre 1 vous permettra de découvrir toute la partie théorique globale nécessaire à la mise en place d'un tel projet.
- Le chapitre 2 quant à lui vous expliquera comment mettre en place la distribution Linux Debian sur laquelle seront installés tous les logiciels et services.
- Le chapitre 3 vous guidera dans l'installation d'un serveur Web Apache avec support php 4 et couplé à un serveur MySql.
- Grâce au chapitre 4 vous apprendrez le fonctionnement et la mise en place d'un serveur DNS primaire, d'un serveur DNS secondaire et la configuration des autres serveurs et clients pour utiliser ces serveurs.
- Le chapitre 5 vous permettra d'installer un serveur DHCP.
- Le chapitre 6 quant à lui vous expliquera comment mettre en place le squelette du réseau: les liens VPNs.
- Vous comprendrez, grâce au chapitre 7, comment mettre en place un Contrôleur Principal de Domaine et un serveur WINS avec le logiciel Samba.
- Le chapitre 8 vous offrira un résumé du travail que vous avez réalisé et vous

préparer à l'installation des services du réseau les plus complexes.
- Le chapitre 9 vous détaillera le fonctionnement et l'installation des protocoles de routage dynamique OSPF et BGP4.
- Grâce au chapitre 10 vous apprendrez à configurer et administrer le logiciel de firewalling sous linux: Netfilter (iptables).
- Le chapitre 11 vous expliquera comment mettre en place la partie la plus complexe du réseau: Radius couplé à Mysql.
- Le chapitre 12 vous premettra de créer le script de démarrage qui réalisera la configuration du serveur à chaque fois que ce dernier est allumé ou redémarré.
- Le chapitre 13 vous présentera le tableau d'attribution des adresses réseau.
- Le chapitre 14 vous permettra de trouver des détails sur des abréviations et termes utilisés dans ce dossier.
- Enfin dans le chapitre 15 vous trouverez des détails sur le licence FDL.

## Ressources nécessaires

Pour bien débuter ce projet de nombreuses ressources vont être nécessaires.

### Configuration matérielle requise

- 1 PC (486 DX minimum, 500Mo de disque, 32Mo de mémoire).
- 2 cartes réseau.
- 1 point d'accès Wireless (pour les utilisateurs mobiles).
- 1 connexion Internet Haut-Débit de type ADSL.

### Configuration Logicielle requise

- 1 distribution Linux Debian (woody: 7 cds + 1 update).
- 1 IP internet fixe ou un domaine de type dyndns.

De plus, il est nécessaire de connaître la plage d'adresses réseau qui vous utiliserez pour la mise en place de votre noeud ou serveur central. Si vous êtes serveur central, il vous appartient de choisir votre plage d'adresses, si vous êtes un utilisateur désirant connecter un noeud à un réseau, adressez-vous au responsable de ce réseau.

## Remerciements

De nombreuses personnes nous ont aidés pour les tests et la mise en place du réseau d'Angers-Wireless qui nous ont permis de réaliser cette documentation.

Thiboult Sylvain (Onyme) – Angers-Wireless : onyme@libertysurf.fr
Malinge Christophe (Lessyv) – Angers-Wireless : lessyv@writeme.com
Adrien (Drien) – Angers-Wireless
Tylski David (Teuxe) – Angers-Wireless
Biton Laurent (Lolo) – ISAIP : cpi02.bitl@isaip.uco.fr
De la Provoté Gwenhaël (Gwe) – ISAIP : gwelapro@yahoo.fr
Ludovic Toinel (Prospere) – Nantes-Wireless : prospere@nantes-wireless.net
Arbey Julien (Psio) – Nantes-Wireless, Brest-Wireless : psio@nantes-wireless.org
Emmanuel GAUTHIER (AngelUS) – Angers-Wireless : e.gauthier@lecolededesign.com

## Sponsors

Nous tenons à remercier les organismes apportant leur soutient au développement de l'association Angers-Wireless.

**Infracom** – Boutique en ligne matériel Radio et réseaux sans fils.
www.onlineinfracom.fr
Belin
F-44160 St Roch
Email : infracom@infracom-france.com
Tél : +33 (2) 40 45 67 67
Fax : +33 (2) 40 45 67 68

**Groupe ISAIP -** Institut Supérieur d' Action Internationale et de Production
www.isaip.uco.fr
18, rue du 8 Mai 1945
BP 80022
49180 Saint-Barthélemy d'Anjou cedex
E-mail : isaip@isaip.uco.fr
Tél. : +33 (0)2 41 96 65 10
Fax : +33 (0)2 41 96 65 11

## Documents de références

La consultation de nombreux sites et documentations en ligne nous ont permis de réaliser cette documentation.

- Alexis de Lattre: Formation Debian GNU/Linux
  http://people.via.ecp.fr/~alexis/formation-linux/formation-linux.html
- Lea-Linux : Site généraliste d'aide linux dans de nombreux domaines
  http://www.lea-linux.org
- Toolinux : Portail sur le monde linux
  http://www.toolinux.com
- Wireless-Lyon: Draft de la solution de Lyon-Wireless
  http://www.wireless-lyon.org
- Linux-France: HOWTO du routage avancé
http://www.linux-france.org/prj/inetdoc/guides/lartc/lartc.dynamic-routing.html
- Liste de diffusion Citrus et Freeradius
  http://lists.cistron.nl/pipermail/
- RFCS : Liste des RFCS abordés dans ce dossier
  http://www.rfc-editor.org/rfc.html
- Nantes-Wireless : Association wireless nantaise
  http://www.nantes-wireless.net

De très nombreux autres sites ont étés consultés, sites découverts grâce à la partie linux du site google : www.google.fr/linux

# Chapitre 1: Présentation détaillée du projet

## A propos de ce chapitre

Ce chapitre présente le fonctionnement global théorique du réseau qui pourra être mis en place grâce à cette documentation.
La finalité de la documentation est de vous permettre d'installer un réseau étendu composé de plusieurs sous-réseaux reliés entre eux par des liens VPN. Sur chacun des points de ce réseau étendu seront proposés différents services.

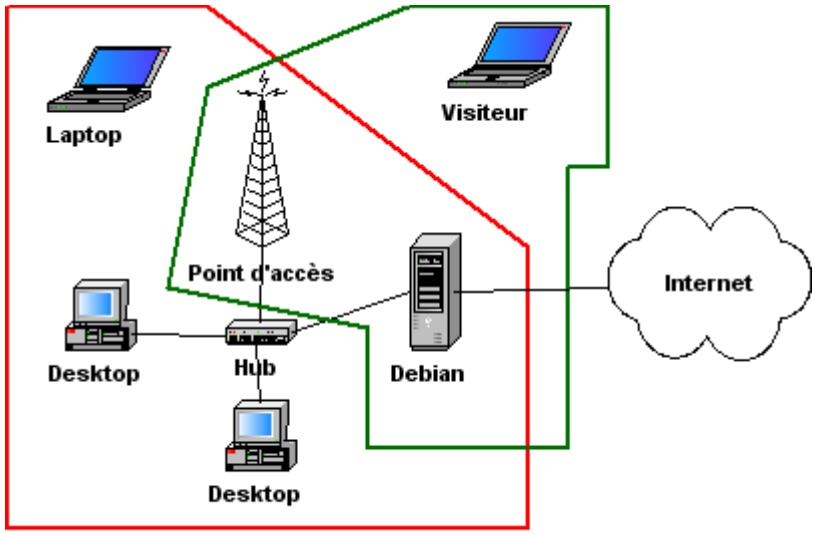
La problématique de la mise en place du réseau peut se résumer de cette façon:

*Permettre à des utilisateurs mobiles (réseaux sans fils) d'accéder de manière sécurisée à des ressources d'un réseau communautaire depuis de nombreux endroits. Les utilisateurs ne devront avoir que peu ou aucune configuration à effectuer sur leur po*ste.

Cette problématique nous aidera lors des choix de mise en place des différentes techniques et logiciels.

## Partie 1: Fonctionnement du réseau local

Dans cette partie nous allons vous présenter le fonctionnement d'un des noeuds (ou node) du réseau. Le serveur central sera lui aussi un node et aura le même fonctionnement avec quelques services en plus.




*Chapitre 1 – Partie 1 - Schéma 1 - Réseau local*

Sur ce schéma, vous nous pouvez voir le fonctionnement "classique" d'une des parties du réseau.
Ici la personne possède un point d'accès compatible Radius, en effet, dans le cas contraire les deux réseaux (rouge et vert) sont physiquement séparés et reliés au serveur par deux cartes réseau différentes.
Le serveur Debian (node ou serveur central) aura trois interfaces réseau, une

interface réseau local, une interface réseau local/étendu et une interface réseau internet.
Sur le réseau local, deux sous réseaux seront créés.
En rouge: un réseau de classe C de type 192.168.1.xxx/24
En vert: un réseau de classe B de type 10.49.xxx.xxx/28

Sur le serveur Debian, un serveur DHCP sera installé distribuant des adresses sur la plage 10.49.xxx.xxx/28.

Les deux réseaux différents permettent une plus grande sécurité, vis-à-vis des utilisateurs non informés, un visiteur se connectant étant automatiquement dirigé vers un réseau différent du réseau local (rouge), il n'aura pas directement accès aux ressources de ce dernier.
L'administrateur du noeud aura donc plusieurs choix:

S'il souhaite une grande confidentialité, il lui suffira de séparer les deux réseaux, s'il souhaite de manière occasionnelle permettre aux deux réseaux de communiquer, il lui suffira de mettre en place une route sur le serveur.

> *Note: l'architecture réseau du serveur dépendra du matériel que possède le propriétaire du noeud, en effet un NAS Radius (explications plus loin dans le dossier) a besoin d'une interface réseau physique qui lui est dédié (sinon les utilisateurs ne seront pas obligés de s'authentifier pour accéder au réseau).*

S'il souhaite que les postes de son réseau local apparaissent sur le réseau étendu, il lui suffira de se connecter à la carte réseau faisant l'authentification RADIUS et de connecter au réseau étendu tous ses postes locaux grâce à des logins / mots de passe.

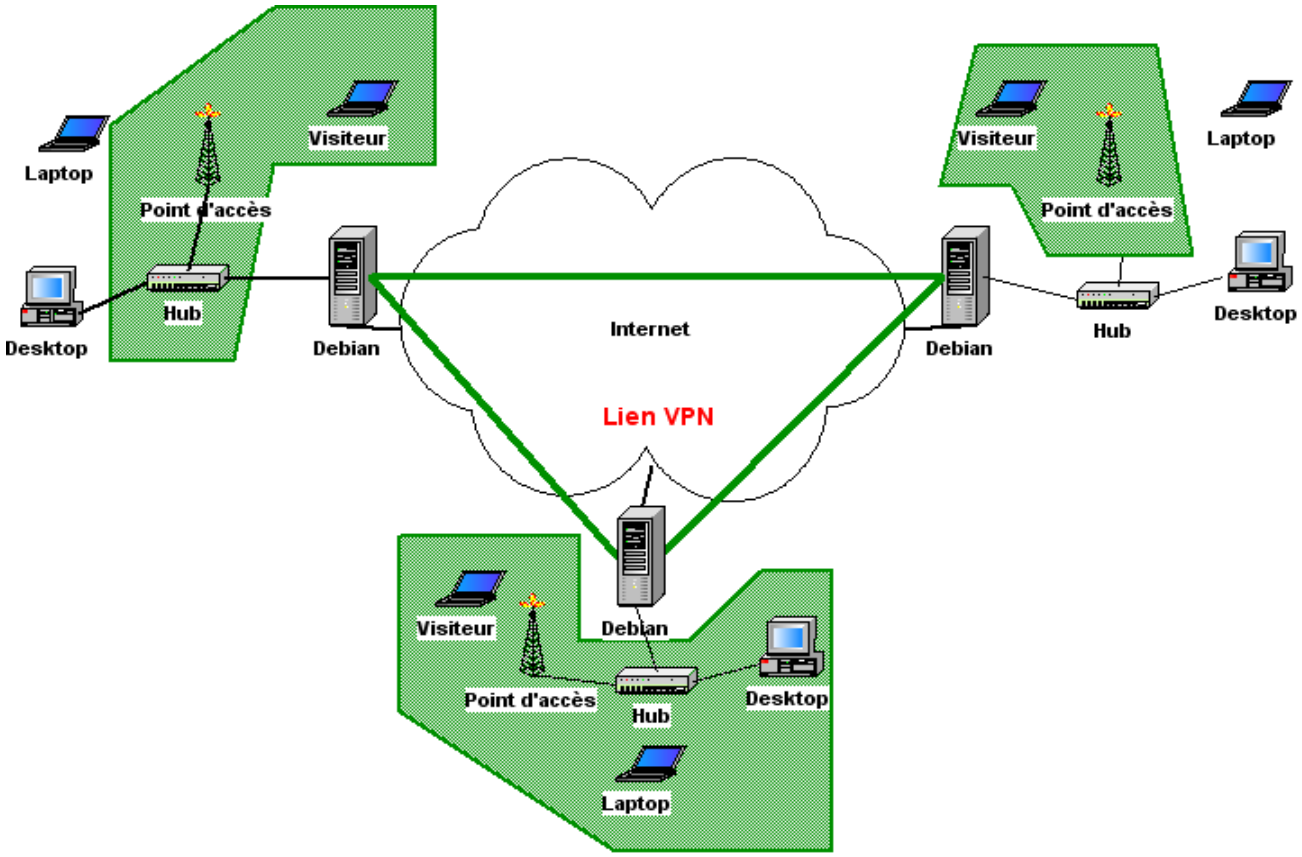
Lors de la création d'un réseau de ce type, il est indispensable de posséder une politique de mot de passe correcte (pas de mot, utiliser la casse et des lettres et chiffres n'ayant aucun rapport entre eux).
Chaque point représenté par le schéma réseau ci-dessus sera appelé un noeud.

Le serveur Debian fera office de master browser (windows) pour les deux réseaux et sera en liaison avec le serveur central, lui même en liaison avec tous les noeuds du réseau. Les postes du réseau local verront donc tous les postes du réseau global, mais ne pourront pas obligatoirement y accéder (explications dans la partie suivante).

L'administrateur du serveur de noeud choisira quels seront les services Internet accessibles aux utilisateurs mobiles connectés à son noeud. Il pourra en effet choisir:

- d'autoriser les utilisateurs mobiles à accéder à n'importe quelle ressource sur Internet.
- d'autoriser seulement quelques services (mail, http, ssh) selon ses choix.

## Partie 2: Fonctionnement du réseau étendu



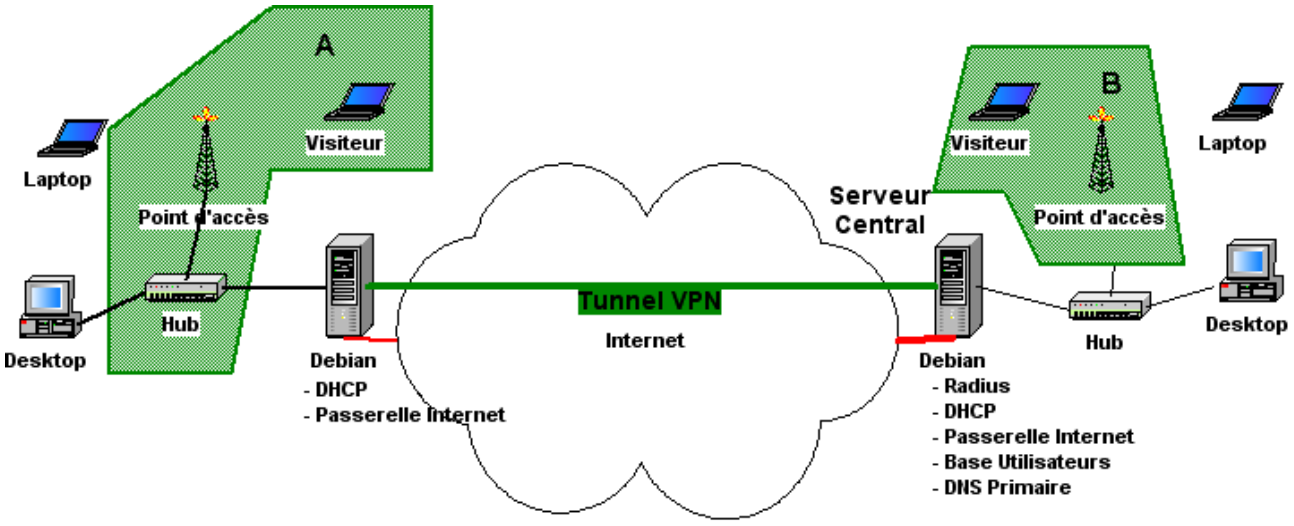

*Chapitre 1 – Partie 2 - Schéma 1 - Réseau étendu*

Les parties colorées en vert correspondent au réseau "étendu" (WLAN), les postes de ce réseau ont une adresse réseau de type 10.49.x.x (voir plus haut) et seront reliées grâce à des liens VPN, ils ne composeront au final qu'un seul grand réseau.

L'administrateur de chaque noeud sera libre de choisir l'adresse de son réseau privé (192.168.1.0/24 par exemple), le réseau étendu ne pourra par défaut pas communiquer avec le réseau privé.
L'administrateur aura la possibilité de migrer tous ses postes sur le réseau étendu (comme le réseau du bas sur le schéma), à condition de posséder les logins / mots de passe nécessaires.

Les personnes situées sur le réseau vert ne pourront pas accéder aux machines des autres noeuds ne faisant pas partie du réseau étendu.
L'accès aux machines des sites locaux depuis un client mobile (une fois authentifié sur le réseau étendu) sera possible si l'administrateur du noeud décide de créer une route réseau sur le serveur de noeud le permettant.

## Partie 3: Scénario




*Chapitre 1 – Partie 3 - Schéma 1 - Scénario*

Nous allons maintenant étudier la connexion sur le réseau A d'un utilisateur mobile, l'administrateur du réseau A a choisi de ne pas partager sa connexion internet.

Le visiteur va se connecter au réseau A , il va s'authentifier auprès du NAS Radius. Ce dernier accédera de manière sécurisée (cryptage) à la base de données du serveur RADIUS central pour vérifier le login et le mot de passe de l'utilisateur.

Si l'utilisateur n'est pas trouvé dans la base du serveur RADIUS central, le NAS Radius essayera tous les serveurs RADIUS (Radius secondaire et serveurs radius des autres communautés) présents dans sa liste de serveurs.

Une fois la connexion validée l'utilisateur pourra accéder aux ressources du réseau. L'administrateur du réseau A ne partageant pas sa connexion internet, il ne pourra pas accéder à internet, seulement à l'intranet et aux différents services du réseau étendu par l'intermédiaire des tunnels VPN.

L'utilisateur pourra accéder à toutes les ressources situées sur tous les postes du réseau local étendu (en vert).
Si l'authentification échoue, l'utilisateur ne pourra accéder à aucune ressource du réseau que ce soit local ou étendu, il ne pourra absolument rien faire.

## Partie 4: Décisions techniques appliquées à Angers-Wireless

Il est important de décider quel sera le statut du serveur que vous allez mettre en place par rapport au réseau. Il y a deux possibilités: le serveur peut soit être le serveur central du réseau, soit être un serveur noeud.

*Note: Pour des raisons de sécurité un des noeuds du réseau aura aussi un rôle de serveur secondaire en cas de panne du serveur principal.*

## Adressage réseau

Par soucis de clarté, chaque département utilise une plage réseau de classe B.

*Note: ceci n'est pas une règle mais plutôt un conseil afin de limiter les doublons dans le cas ou tous les réseaux seraient reliés entre eux.*

Exemples d'adressage:

Maine et Loire : 10.49.0.0
Loire Atlantique: 10.44.0.0
Finistère: 10.29.0.0

Ce type d'adressage autorise plus de 65000 postes sur le réseau, il est de plus nécessaire (pour des raisons de routage) de subdiviser le réseau en autant de sous réseau que de noeuds.
Chaque noeud se verra allouer 16 adresses IP pour son réseau.
Chaque noeud aura un masque de 255.255.255.240

## Détails sur l'adressage

Voici un tableau récapitulatif de l'adressage réseau.

| Nom | ad.réseau | Masque | Broadcast | IP Serveur / Node | Plage IP |
|---|---|---|---|---|---|
| Serv. Central | 10.49.1.0 | 255.255.255.240 | 10.49.1.15 | 10.49.1.1 | 2 à 14 |
| Noeud 1 | 10.49.1.16 | 255.255.255.240 | 10.49.1.31 | 10.49.1.17 | 18 à 30 |
| Noeud 2 | 10.49.1.32 | 255.255.255.240 | 10.49.1.47 | 10.49.1.33 | 34 à 46 |
| ... | ... | ... | ... | ... | ... |

*Note: Chaque serveur (central, node ou secondaire) utilisera la première adresse IP du réseau. L' "IP tunnel" correspond à l'IP utilisé par le logiciel vtun (VPN) pour communiquer à travers le lien VPN, plus d'information dans la partie du dossier consacrée à ce logiciel.*

*Note 2: Un tableau plus complet est disponible en annexe.*

## Serveur central

Le serveur central a pour rôle de centraliser les informations importantes du domaine, principalement les logins/pass des utilisateurs du réseau. Il ne doit y avoir qu'un seul serveur central par réseau. Le serveur central fera de plus office de serveur DNS primaire, permettant notamment aux utilisateurs extérieurs d'accéder aux services internet de chaque noeud, grâce à une adresse dns du type noeud.serveur_principal.net

*Note: L'accès aux ressources des nodes se fera grâce au serveur DNS, qui renverra aux utilisateurs l'adresse IP "internet" du node demandé.*

*Note 2: Pour que les requêtes DNS puissent êtres résolues depuis Internet, il faut s'inscrire auprès d'un registar (payant), les DNS gratuits de types dyndns ne permettent pas ce type de résolution.*

Le serveur central servira de lien entre les différentes communautés, il sera le point par ou passeront toutes les requêtes vers les autres communautés.
Services installés sur le serveur secondaire:

- Serveur DNS primaire
- Serveur Radius primaire
- Tunnels vers les autres communautés
- Contrôleur Principal de Domaine
- Serveur WINS
- Tunnels intra-association
- Routage Dynamique Zebra / OSPF
- Firewall
- DHCP (dans le cas ou le serveur central fait aussi office de noeud)

## Serveur secondaire

Il y aura un serveur secondaire sur le réseau, le serveur secondaire aura le même statut qu'un noeud avec en plus des services lui permettant de prendre le rôle de serveur principal en cas de panne de ce dernier.
Services installés sur le serveur secondaire:

- Serveur DNS secondaire
- Serveur Radius secondaire
- Tunnels secondaires vers les autres communautés
- Tunnels intra-association
- Routage dynamique Zebra / OSPF
- Firewall
- DHCP (dans le cas ou le serveur central fait aussi office de noeud)

## Noeuds

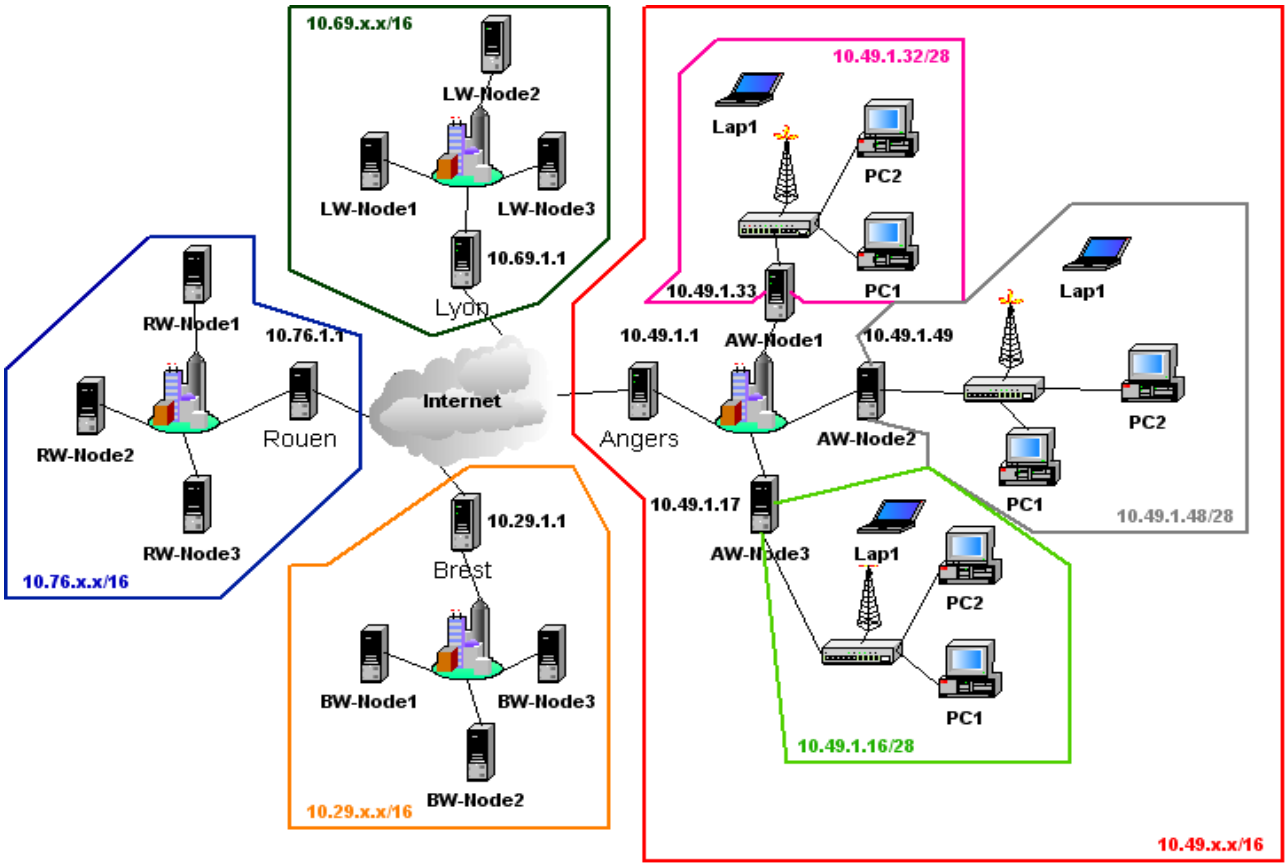
Il peut y avoir un nombre élevé de noeuds, un noeud aura pour tâche de réaliser, le lien entre le réseau local et le réseau étendu.

Lors de ce dossier, nous allons suivre un ordre logique dans l'installation des différents logiciels, nous commencerons par l'installation du réseau en lui même puis nous aborderons les différents services qui peuvent êtres proposés par ce dernier.

Lors de l'ajout d'un noeud, le propriétaire du nouveau noeud devra contacter tous les propriétaires des autres noeuds pour pouvoir échanger les mots de passe et les informations de configuration (principalement vtun).
Services installés sur les noeuds:

- NAS Radius
- Tunnels intra-association
- Routage dynamique Zebra / OSPF
- Firewall
- DHCP (dans le cas ou le serveur central fait aussi office de noeud.

## Schématisation réseau global




*Chapitre 1 – Partie 4 - Schéma 1 – Réseau global*

*Note: Les noms des villes sur ce schéma ne sont là qu'à titre indicatif et ne sont pas représentatifs de la réalité.*

Voici le schéma global détaillé du futur réseau, ce graphique à pour objectif de schématiser les décisions techniques précédentes.
Vous pouvez voir les communautés de différentes villes, chacune ayant une adresse IP de type 10.xxx.xxx.xxx/16 .
Sur les réseaux de ces communautés seront placés plusieurs nodes (RW-Node1, BW-Node2, ...) reliés au serveur central (10.xxx.1.1) et reliés entre eux par l'intermédiaire d'internet.

*Note: la solution présentée est celle d'Angers, chaque communauté est libre de développer son propre système avec différents services proposés sur le réseau, la seule partie commune étant l'identification des utilisateurs.*

Sur chaque node seront reliés des postes clients (PC1, Lap1, ...) de manière filaire ou wireless (l'authentification sera abordée par la suite).
Chaque node aura son propre sous-réseau et chaque poste d'un sous réseau devra être en mesure d'échanger des informations avec d'autres postes d'un autre sous réseau (le filtrage des informations pouvant êtres échangées sera abordé par la suite).

# Chapitre 2: Installation de Linux – Debian

## A propos de ce chapitre

Dans ce chapitre vous allez découvrir comment installer Linux sur votre serveur (noeud ou central), grâce à un guide réalisé par Alexis De Lattre. Puis vous verrez comment configurer les services de bases et comment utiliser ssh.

## Partie 1: Installation de Debian Linux

Nous allons maintenant débuter l'installation de votre serveur. Il existe un très bon guide expliquant comment installer une distribution Debian Woddy. L'objectif de cette documentation n'étant pas de détailler l'installation d'une distribution Linux, nous vous conseillons de vous référer au lien ci-dessous pour la mise en place de votre serveur.

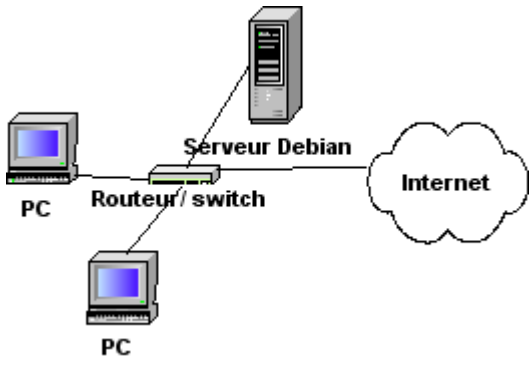
Si vous êtes capables d'installer une distribution Debian par vous même sachez que pour pouvoir suivre les indications de ce dossier votre serveur doit être en mesure d'accéder à internet, avoir ses interfaces réseau de configurées, et être administrable par ssh.

Si vous suivez le guide, vous pouvez vous arrêter au chapitre 21.

Le guide est disponible à cette adresse:
http://www.via.ecp.fr/~alexis/formation-linux/formation-linux.html

Votre serveur devrait donc maintenant être configuré correctement.

## Réseau avec routeur

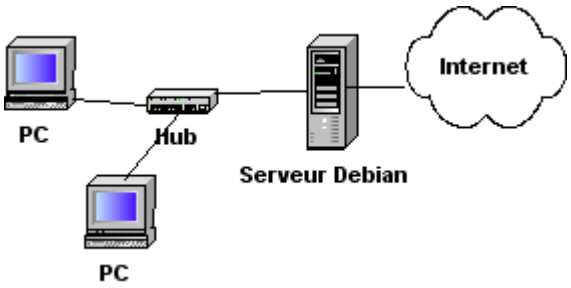



*Chapitre 2 – Partie 1 - Schéma 1 - Réseau avec routeur*

Si votre réseau comporte un routeur, il est nécessaire de définir votre serveur en tant que DMZ (option présente dans la majorité des routeurs matériel). Si votre routeur faisait du mapping de port, ne vous inquiétez pas, le serveur peut très facilement faire du forwarder les ports. Nous aborderons ce point plus loin dans ce dossier lors de la configuration de Netfilter (iptables). Vérifiez que votre connexion internet est fonctionnelle depuis le serveur,

grâce à un ping par exemple.

### Réseau sans routeur




*Chapitre 2 – Partie 1 - Schéma 2 - Réseau sans routeur*

Si vous possédez un réseau et que votre serveur doit faire office de routeur internet, il peut être intéressant pour ne pas immobiliser le réseau durant votre installation, de partager la connexion Internet

Exécutez les commandes suivantes:

```
1:# echo 1 > /proc/sys/net/ipv4/ip_forward
2:# iptables -A POSTROUTING -t nat -o ppp0 -j MASQUERADE
3:# apt-get install wget
```

Ligne 1: Active le forwarding du noyau Linux
Ligne 2: Cache les machines forwardées par le firewall
Ligne 3: Installation du logiciel wget permettant de télécharger des fichiers sur Internet

*Note: remplacez ppp0 par le nom de votre interface réseau connectée à internet, généralement ppp0 pour les connections modems classiques*
*Vous trouverez plus de détails sur iptables et sa configuration dans la partie qui lui est consacrée dans ce dossier.*

Cette configuration est très permissive et n'est que provisoire, le temps que vous finissiez l'installation du réseau.

## Partie 2: Installation des sources et compilation du noyau

Nous aurons probablement par la suite à compiler des applications, et il peut être nécessaire d'installer les sources du noyau linux.
De plus nous allons ajouter le support PPP qui sera nécessaire par le suite.

```
1:# apt-get install kernel-source-2.4.18
2:# apt-get install bzip2 libncurses5-dev
3:# cd /usr/src/
4:# tar xvjf kernel-source-version.tar.bz2
5:#  ln -sfn kernel-source-2.4.18 linux
6:# cd linux/ ; ls
```

Ligne 1: Installation des sources depuis le cdrom ou internet
Ligne 2: Installation du package libncurses5
Ligne 3: Déplacement dans le repertoire contenant l'archive installée
Ligne 4: Décompression des sources
Ligne 5: Création d'un lien symbolique appelé "linux" vers le répertoire décompressé
Ligne 6: Vérifiez la présence de fichiers dans le répertoire linux

Nous allons maintenant patcher le noyau pour le ajouter le support PPP

```
1:# cvs -z3 -d:pserver:cvs@cvs.samba.org/cvsroot co ppp
2:# cd ppp/linux/mppe
3:# sh mppeinstall.sh /usr/src/linux
4:# cd /usr/src/linux ; make menuconfig
```

Ligne 1: Téléchargement en cvs de ppp
Ligne 2: déplacement dans le repertoire mppe de ppp
Ligne 3: Patchage du kernel pour utiliser mppe
Ligne 4: Nous entrons dans l'interface de configuration du noyau

Configurer le noyau selon vos choix ou reportez vous à une aide sur le sujet. Mais ajoutez tous les supports pour ppp, rubrique Network device support puis mettez en built-in (*) tout ce qui concerne PPP.

Pour la compilation et la mise en place du noyau je vous conseille de vous référer à la documentation d'Alexis de Lattre, chapitre 9.

## Partie 3: L'administration via ssh

Le logiciel OpenSSH permet de déporter de manière sécurisée une console sur un autre poste du réseau. Grâce à ce logiciel, vous pouvez réaliser toutes sortes de tâches d'administration.

*Note: Le serveur ssh à été automatiquement installé durant la mise en place du serveur (au moment de l'étape dselect).*

## Fichiers

Exécutable serveur:
/etc/init.d/ssh
Paramètres de l'exécutable:

| | |
|---|---|
| start | *Démarre le serveur ssh.* |
| stop | *Arrête le serveur ssh.* |
| reload | *Recharge le fichier de configuration du serveur ssh.* |
| Force-reload | *Recharge le fichier de configuration du serveur ssh sans prendre en compte les erreurs.* |
| restart | *Redémarre le serveur ssh* |

Exemple d'utilisation, démarrage du serveur:

```
1:# /etc/init.d/ssh start
```

Fichier contenant notamment le log des connections: /var/log/auth.log

```
1:# more /var/log/auth.log | grep ssh
```

Ligne 1: affichage du log de connexions ssh

Fichier de configuration du serveur:
/etc/ssh/sshd_config
Paramètres principaux:

| | | |
|---|---|---|
| Port | 22 | 22 est le port par défaut pour le ssh, pour écouter sur plusieurs ports en même temps, il suffit de mettre cette ligne autant de fois que nécessaire |
| Protocol | 2 | *N'accepte que les clients utilisant la version 2 du protocole (plus sécurisé)* |
| | 1 | *N'accepte que les clients utilisant la version 1 du protocole* |
| | 2,1 | *Accepte toutes les versions du protocole* |
| PermitRootLogin | yes | *Autorise l'utilisateur root à se logguer en ssh* |
| | no | *N'autorise pas l'utilisateur root à se logguer en ssh. L'utilisateur pourra se connecter par la suite en root grâce à la commande su, technique plus sécurisée car la personne souhaitant se connecter doit au moins connaître 2 mots de passes.* |
| X11Forwarding | yes | *Permet de faire de l'affichage distant grâce à du ssh* |
| | no | |
| ListenAddress | 192.168.0.1 | *N'autoriser seulement des connections venant de cette adresse ip, pour tout autoriser ne pas mettre cette ligne.* |
| AllowUsers | root, toto, etc | *N'autoriser que les connections de certains utilisateurs* |
| AllowGroups | admin, toto, etc | *N'autoriser que les connections de certains groupes d'utilisateurs* |

Un exemple de fichier de configuration fonctionnel est disponible à la fin de ce dossier.

## Utilisation de ssh

L'utilisation de ssh est extrêmement aisée, il suffit depuis un poste distant de taper la commande suivante:

```
1:# ssh root@adresse_ip_serveur_ou_nom_dns
```

Ligne 1: connexion en ssh en tant que root à un serveur ssh
*Le mot de passe sera demandé par la suite, vous pouvez vous connecter avec n'importe quel utilisateur présent sur le serveur.*

Vous vous retrouvez ensuite sur un shell distant se comportant de la même manière qu'un shell local.

## Partie 4: Configuration réseau

Votre serveur est maintenant installé et configuré pour votre réseau local, avec une adresse IP de votre choix. Nous allons l'adapter au réseau étendu.
Nous allons lui attribuer une adresse IP selon les exemples évoqués précédemment, pour plus de détails vous trouverez un tableau d'adressage complet à la fin de ce document.

Nous allons tout d'abord commencer par fixer la redirection automatique du noyau.
Fichier /etc/network/options

```
ip_forwarding=yes       #active la redirection du noyau
syncookies=no           #protège votre poste contre les attaques de type syn-flood*
```

Nous allons définir l'IP fixe de votre deuxième carte réseau.

Fichier /etc/network/interfaces

```
auto lo
iface lo inet loopback
auto eth0, eth1
iface eth0 inet static
        address 192.168.1.1             #adresse IP de votre serveur sur le réseau local
        netmask 255.255.255.0           #Masque de sous réseau de votre réseau local

iface eth0:0 inet static
        address 10.49.1.1               #adresse IP de votre serveur sur le réseau étendu
        netmask 255.255.255.240         #Masque de sous réseau de votre node
        network 10.49.1.0               #Adresse réseau de votre node
        broadcast 10.49.1.15            #Adresse de broadcast de votre node

iface eth1 inet static
        address 10.0.0.1          #adresse IP de votre serveur sur le réseau étendu
        netmask 255.255.255.128   #Masque de sous réseau de votre node
        network 10.0.0.0          #Adresse réseau de votre node
```

# Chapitre 3: Installation et mise en place du serveur Web

## A propos de ce chapitre

Dans ce chapitre vous découvrirez comment installer un serveur Web Apache et comment le faire fonctionner avec le module php et comment installer le serveur de base de données Mysql.
Apache va être utilisé, pour mettre en place le serveur intranet et le logiciel php qui nous servira à administrer le serveur. Nous installerons de plus un logiciel php permettant d'administrer les bases de données mysql: phpMyAdmin.

## Partie 1: Installation des logiciels

Nous allons maintenant installer tous les packages nécessaires.

```
1:# apt-get install apache
```

Ligne 1: Installation du serveur Web Apache

```
1:# apt-get install php4 php4-mysql php4-cgi
```

Ligne 1: Installation du support php4 pour apache

Lors de l'installation de php4 plusieurs questions vous seront posées:

*Do you want me to run the apacheconfig script now ? [y/N] :* répondez **y**
*Save these changes to the configuration files ? [Y/n] :* répondez **y**
*Restart Apache now ? [Y/n] :* répondez **n**
*extension=mysql.so*
*Do you want me to add it now [Y/n] ?:* répondez **y**

```
1:# apt-get install mysql-server
```

Ligne 1: Installation du serveur mysql

*Sécurity and update notice: appuyez sur **entrée***
*Should I remove everything ... : répondez **yes***
*Should MySQL start on boot:répondez **yes***

## Partie 2: Configuration du serveur Web

Une fois tous les serveurs installés il est nécessaire de les configurer.

Les détails de chaque option du fichier de configuration d'Apache ne seront pas

détaillés dans cette partie.

Modifiez les parties suivantes du fichier /etc/apache/httpd.conf

| Ligne | Variable | Valeur | Commentaires |
|---|---|---|---|
| 310 | `ServerName` | Nom_DNS | Mettez ici le nom DNS de votre serveur. |
| 358 | `AllowOverride` | All | Mettre en place des restrictions d'accès |

Toujours dans le fichier de configuration, dé-commentez les lignes suivantes:

*Ligne 241 :=> LoadModule php4_module /usr/lib/apache/1.3/libphp4.so*
*Ligne 757 :=> AddType application/x-httpd-php .php*

La configuration étant maintenant finie, il est nécéssaire de redémarrer apache

```
1:# /etc/init.d/apache restart
```

Ligne 1: Redémarrage du serveur apache

## Partie 3: Installation de PhpMyAdmin

Nous allons maintenant installer PhpMyAdmin, ce logiciel permet de réaliser de l'administration de bases de données mysql, de plus il nous permettra de tester le bon fonctionnement de notre serveur.

```
1:# cd /var/www/
2:# wget
http://keihanna.dl.sourceforge.net/sourceforge/phpmyadmin/phpmyadmin-2.5.4-php.tar.gz
3:# tar xfvz phpmyadmin-2.5.4-php.tar.gz
4:# mysqladmin -u root password NouveauMotDePasse
```

Ligne 1: Se déplacer dans le répertoire par défaut ou sont situées les pages internet
Ligne 2: Récupérer l'archive du logiciel.
Ligne 3: Décompresser le logiciel
Ligne 4: Changer du mot de passe mysql de l'utilisateur root

Nous allons maintenant configurer phpMyAdmin
Éditez le fichier /var/www/phpMyAdmin-2.5.4/config.inc.php
Ligne 79 :=> remplacez 'config' par 'http'
Ligne 80 :=> remplacez 'root' par ''

Vous pouvez tester l'installation grâce à un navigateur web en tapant l'adresse suivante: http://IP_de_Votre_Serveur/phpMyAdmin-2.5.4/index.php

Si tout se passe bien, un boite de dialogue apparaîtra vous demandant votre login et mot de passe.

# Chapitre 4: Installation et mise en place du DNS

## A propos de ce chapitre

Le service DNS (Domain Name Service) est un service qui va nous permettre de réaliser le lien entre des adresses IP et des noms de domaine et inversement.
Vous découvrirez dans ce chapitre le fonctionnement des serveurs DNS, ainsi que l'installation d'un serveur DNS primaire et d'un serveur DNS secondaire sur un réseau. Enfin vous découvrirez comment configurer les noeuds pour fonctionner avec ces serveurs DNS.

## Partie 1: Rappels

Le service DNS à été créé pour simplifier la vie des utilisateurs, en effet il est beaucoup plus simple de retenir angers-wireless.net que son adresse IP 89.125.23.9. Un utilisateur cherchant à visiter le site internet aura juste à taper www.angers-wireless.net dans son navigateur favori pour accéder au serveur web d'Angers-wireless.

Mais pour que cette redirection fonctionne, il faut bien que la relation entre le nom et l'adresse IP soit inscrite quelque part, en effet le navigateur ne s'adresse pas au nom inscrit dans la barre d'adresse mais à l'adresse IP correspondante.
Les serveurs DNS fonctionnent selon un système d'arborescence, les demandes s'adressent tout d'abord à un des serveurs " . " (" point ") et la requête est renvoyée sur des domaines inférieurs jusqu'à aboutissement de la requête.

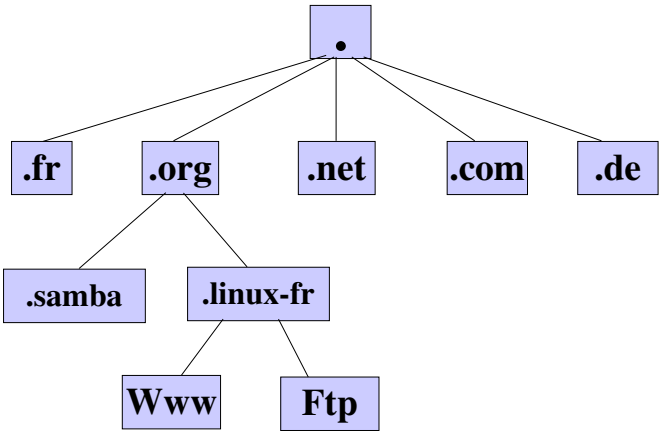



*Schéma 7.2.1 – 1 Arborescence DNS*

Sur internet il y a donc des serveurs spécialisés dans la résolution de ces requêtes.
Les requêtes fonctionnent de la manière suivante:

- Un utilisateur cherche à accéder au site www.linux-fr.org, il entre donc l'adresse dans son navigateur web préféré.
- Le navigateur web va s'adresser au plus haut serveur DNS dans l'arborescence, le serveur "." ("point"), il y en a 14 en tout sur internet.
- Le serveur "." contient la liste des adresses des serveurs DNS de second niveau (les net, org, com, fr, ru, ...), il va donc renvoyer le navigateur vers le serveur

DNS correspondant (le .org dans notre cas)
- Le navigateur web s'adresse donc au serveur DNS de second niveau (le .org) et va lui demander l'adresse IP du serveur linux-fr.org
- Le navigateur web possède maintenant l'adresse IP du serveur web linux-fr.org, nous pouvons donc maintenant visiter notre site web favori

Toutes les requêtes sur internet ne s'adressent pas tout le temps au serveurs "point", en effet les DNS possèdent des caches, dans lesquels sont stockés les requêtes ayant déjà étés réalisées. Votre fournisseur d'accès vous fourni un ou plusieurs DNS, vous permettant de réaliser les translations, si le DNS de votre fournisseur connais l'adresse IP d'un domaine demandé par un utilisateur, il n'ira pas s'adresser à un serveur "point", mais cherchera les informations dans son cache.

## Partie 2: Installation de bind9

L'installation du service DNS est extrêmement simple, il suffit de taper la commande suivante:

```
1:# apt-get install bind9
```

Ligne 1: Installation du serveur DNS

## Partie 3: Configuration du serveur DNS primaire

Pour la communauté française le domaine fw sera utilisée (france-wireless), chaque ville ayant un domaine de type ville.fw. Les exemples précédant ne seront pas utilisés. Leur objectif était de vous permettre de comprendre le fonctionnement du service DNS.

Le répertoire /var/cache/bind/ est destiné à accueillir les fichiers de zone des DNS primaires et secondaires.

Le serveur DNS va retenir dans son cache, pendant une certaine période, les correspondances IP <-> DNS qui ont été demandées par les clients.

Si le serveur n'est pas en mesure de répondre aux demandes des clients (s'il ne possède pas dans son fichier de configuration ou dans son cache la correspondance IP <-> DNS), il transmettra automatiquement la demandes à d'autres serveurs DNS, dans notre cas les DNS du FAI.

Fichier /etc/bind/named.conf

```
options {
        directory "/var/cache/bind";

        forwarders {
                xxx.xxx.xxx.xxx; #adresse IP des DNS de
         };                          votre provider

        auth-nxdomain no;
};
```

Il faudra ensuite modifier le fichier /etc/resolv.conf pour attribuer l'ordre dans lequel seront interrogés les serveurs DNS. Nous avons placé notre serveur DNS en tête de liste.

```
search angers.fw
nameserver 127.0.0.1
nameserver 10.49.1.1
```

Comme nous avons un nom de domaine "angers.fw", nous pouvons utiliser ce serveur en tant que DNS pour tout le réseau. A noter que d'autres serveurs (dit "secondaire") pourront être présent, pour prendre le relais en cas de panne du serveur principal.

Nous avons ajouté les lignes suivantes à la fin du fichier named.conf:

```
zone "angers.fw" {
        type master;
        file "angers.fw.zone";
};
```

Détails:

- *angers.fw* est le nom de domaine pour lequel votre serveur sera primaire,

- *angers.fw* désigne le fichier "/var/cache/bind/angers.fw.zone" où seront stockés les enregistrements de la zone.

Pour vérifier si le fichier named.conf est bien configuré, il est nécessaire d'éxécuter la commande:

```
1:# named-checkconf
```

Ligne 1: Vérification de la configuration

Fichier de zone
**/var/cache/bind/angers.fw.zone**

Organisation d'un fichier de zone

```
@       IN      SOA     domaine.net.    {       ; 1: Début enregistrement SOA
        xxxxx           ; 2: Serial
        xxxxx           ; 3: Refresh
        xxxxx           ; 4: Retry
        xxxxx           ; 5: Expire
        xxxxx           ; 6: TTL
}
domaine.net.    IN      NS      ns0                     ; 7: Enregistrement(s) NS
AWServer        IN      A       xxx.xxx.xxx.xxx         ; 8: Enregistremens A
Onyme           IN      A       xxx.xxx.xxx.xxx         ; 8: Enregistrements A
www             IN      CNAME   AWServer                ; 9: Enregistrements CNAME
```

Ligne 1: Début de l'enregistrement SOA (Start of a zone of Authority), cet enregistrement permet de définir le nom du serveur DNS primaire
Ligne 2: Le serial est le numéro d'enregistrement de la zone DNS, il permet au serveur DNS de savoir si le fichier a été modifié. Pour valider l'enregistrement, il est nécéssaire de modifier le serial à chaque modification sinon lors du reload, le serveur ne rechargera pas le fichier de configuration.
Ligne 3: Toutes les xxx secondes le serveur DNS secondaire vérifiera auprès du serveur DNS primaire si le fichier de zone à été modifié.
Ligne 4: Toutes les xxx secondes, si le serveur DNS primaire n'a pas répondu à une requète Refresh (ligne 3), le serveur DNS secondaire retentera de se connecter pour mettre à jour ses enregistrements
Ligne 5: Si les serveurs secondaires n'arrivent pas à contacter le serveur primaire, au bout de xxx secondes, ils considèrent que les informations de leur cache sont invalides.
Ligne 6: Un lien DNS <-> IP n'est valide dans le cache que pendant xxx secondes
Ligne 7: ns0, Nom du serveur primaire de la zone NS
Ligne 8: Permet de faire la correspondance entre le nom DNS et l'adresse IP
Ligne 9: L'enregistrement CNAME permet de créer des sous-domaines pointant vers des enregistrements A

*Note: L'indicateur de commentaires dans les fichiers de zone est le ";" et non le "#"*

Exemple de fichier de zone

```
@       IN      SOA     angers.fw.      {
        2003111803      ; Serial
        86400           ; Refresh de 86400 secondes (1 jour)
        300             ; Retry toutes les 300 secondes
        2592000         ; Expire au bout de 2592000 secondes (1 mois)
        86400           ; TTL de 86400 secondes
}
angers.fw.      IN      NS      ns0             ; ns0 serveur primaire de la zone
AWServer        IN      A       10.49.1.1       ; Adresse ip du serveur AWServer
Onyme           IN      A       10.49.1.49      ; Adresse ip du node Onyme
www             IN      CNAME   Onyme           ; www pointe vers Onyme
```

Détails:
Pings réalisés depuis des machines internes ayant seulement notre DNS primaire de renseigné.

| Requête (ping ou autre) | Ip en réponse | Commentaires |
|---|---|---|
| Angers.fw | 10.49.1.1 | Renvoie l'adresse IP du serveur DNS |
| AWServer.angers.fw | 10.49.1.1 | " |
| Onyme.angers.fw | 10.49.1.49 | Adresse IP locale du noeud Onyme |
| www.Onyme.angers.fw | 10.49.1.49 | Adresse IP locale du noeud Onyme |

Une fois la configuration du fichier de zone effectuée, il est nécessaire de vérifier la syntaxe de ce fichier:

```
1:# named-checkzone angers.fw /var/cache/bind/angers.fw
zone angers.fw/IN: loaded serial 2003111803
OK
```

Ligne 1: Vérification de la syntaxe de la zone angers.fw

La commande affiche bien que la zone du domaine angers.fw ayant le numéro de série 2003111803 est correcte (OK).

Il est nécessaire après chaque modification, de recharger le fichier de configuration.

```
1:# /etc/init.d/bind9 reload
```

Ligne 1: rechargement des fichiers de zone DNS
*Note: Si vous faites un "restart" à la place du "reload", le cache du serveur DNS se videra !*

## Partie 4: Configuration du serveur DNS secondaire

Cette partie la est relativement aisée à mettre en oeuvre, commencez par installer bind9 comme décrit plus haut.
Puis modifiez les fichiers de la manière suivante.

Fichier /etc/bind/named.conf

```
options {
        directory "/var/cache/bind";
        notify no; #desactive l'envoi de message aux
                        #serveurs esclaves
        forward only;
        forwarders {
                xxx.xxx.xxx.xxx; #adresse IP des DNS de
         };                          votre provider


        auth-nxdomain no;
};
```

Puis ajoutez à la fin du fichier named.conf

```
zone "angers.fw" {
        type slave;
        file "angers.fw.zone";
        masters { 10.49.1.1 };
};
```

Ensuite faites un "reload" pour recharger le fichier de configuration.

```
1:# /etc/init.d/bind9 reload
```

Ligne 1: rechargement des fichiers de zone DNS

*Note: Si vous faites un "restart" à la place du "reload", le cache du serveur DNS se videra !*

## Partie 5: Configuration des noeuds pour utiliser le serveur

Nous allons maintenant configurer les noeuds pour fonctionner avec le serveurs central et les serveurs secondaires au cas ou ce dernier tombe en panne.

Il faudra modifier le fichier /etc/resolv.conf pour attribuer l'ordre dans lequel seront interrogé les serveurs DNS. Nous avons placé notre serveur DNS en tête de liste.

```
search angers.fw
nameserver 10.49.1.1 #Dns primaire
nameserver 10.49.1.81 #Dns secondaire
#vous pouvez rajouter d'autres Dns secondaires pour plus
#de sécurité
```

# Chapitre 5: Installation et mise en place du DHCP

## A propos de ce chapitre

Dans ce chapitre vous découvrirez comment installer et configurer un serveur DHCP. Le DHCP est un protocole permettant d'attribuer automatiquement des adresses IP aux postes du réseau. Dans notre cas il nous servira à attribuer les adresses qui ne seront utilisées pour la connexion d'un utilisateur au réseau.

## Partie 1: Installation du serveur DHCP

```
1:# apt-get install dhcp
```

Ligne 1: Installation du serveur dhcp

## Partie 2: Configuration du serveur DHCP

Exécutable serveur:
/etc/init.d/dhcpd
Paramètres de l'exécutable:

| | |
|---|---|
| start | *Démarre le serveur dhcp.* |
| stop | *Arrête le serveur dhcp.* |
| status | *Affiche l'état du serveur dhcp.* |
| condrestart | *Redémarre le serveur.* |
| restart | *Redémarre le serveur dhcp.* |

Exemple d'utilisation, démarrage du serveur:

```
1:# /etc/init.d/dhcpd start
```

Ligne 1: Démarrage du serveur dhcp

Il est nécessaire de rappeler quelles sont les différentes adresses réseau qui ont étés décidées précédemment. Nous garderons le même exemple durant tout le dossier.

*Extrait du tableau d'adressage*

| Nom | ad.réseau | Masque | Broadcast | IP Serveur / Node | Plage IP | IP tunnel |
|---|---|---|---|---|---|---|
| Serv. Central | 10.49.1.0 | 255.255.255.240 | 10.49.1.15 | 10.49.1.1 | 2 à 14 | 10.49.0.10 |

Fichier de configuration du serveur dhcp:
/etc/dhcpd.conf

```
ddns-update-style none;
subnet 1.0.0.0 netmask 255.255.255.128  # adresse réseau et masque de sous réseau.
{
option routers 10.0.0.1;                #adresse du routeur.
option subnet-mask 255.255.255.128;         #masque de sous réseau.
Range 10.0.0.2 10.0.0.100;              # plage d'attribution automatique des adresses IP.
default-lease-time 21600;               #temps par défaut d'attribution d'une IP à une adresse MAC
max-lease-time 43200;                   #temps max d'attribution d'une IP à une adresse MAC
}
```

Les clients qui se connecteront maintenant se verront automatiquement attribuer une adresse IP sur la plage définie ci dessus..

# Chapitre 6: Installation et mise en place de Vtun

## A propos de ce chapitre

Dans ce chapitre vous apprendrez à installer, à configurer et à utiliser le logiciel Vtun. Ce logiciel permet de réaliser du tunneling IP (VPN) à travers internet.
Un VPN (Vitrual Private Network) est un tunnel créé entre 2 réseaux, permettant à ces deux réseaux d'échanger des données directement. Habituellement cryptés et fiables les VPN sont des plus en plus utilisés dans les entreprises pour relier plusieurs sites entre eux, ils sont une alternative financièrement très intéressantes aux lignes privées. Un VPN peux être comparé à un câble reliant à travers internet deux réseaux ou 2 postes.
Le logiciel Vtun a été choisi, notamment pour sa facilité de mise en oeuvre et sa sécurité (cryptage 128 bits), il a une réputation de fiabilité et de performance. De plus vtun peut fonctionner avec des noms de domaines (à la place d'adresses IP) ce qui est extrêmement pratique lorsque les noeuds ne possèdent pas d'IP internet fixe.

## Partie 1: Installation de Vtun

```
1:# mkdir -p /dev/net/
2:# mknod /dev/net/tun c 10 200
3:# mkdir -p /dev/misc/net
4:# ln -sf /dev/net/tun /dev/misc/net/tun
5:# apt-get install vtun
```

Ligne 1: Création du répertoire /dev/net/
Ligne 2: Création d'un device
Ligne 3: Création du répertoire /dev/misc/net
Ligne 4: Création d'un lien symbolique de /dev/net/tun vers /dev/misc/net/tun
Ligne 5: Installation de Vtun

# Partie 2: Configuration

## Fichiers

Fichier /etc/vtund.conf

```
#partie commune
option {
  port 5000;        #Port à utiliser
  syslog daemon; #Fichier de logs à utiliser
}
default {
  type tun;         #Connexion de type tunnel
  proto tcp;        #Utilise TCP
  comp lzo:1;       #Encrypte le données (128 bits)
  keepalive yes;    #Garde la connection active
  stat yes;         # Statistiques d'utilisation
  speed 0;          # Pas de limite de vitesse, pour limiter mettre une valeur en Kb/s
}
#coté client
NomDuLien {
  pass MotDePasse;        #Le mot de passe doit être commun au client et au serveur
  persist yes;            #Garder le lien en activité
  device tun0;            #Périhpérique vitruel à utiliser pour le tunnel
  up {
    ifconfig "%% xxx.xxx.xxx.xxx pointopoint xxx.xxx.xxx.xxx";
    route "add -net xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx gw xxx.xxx.xxx.xxx";
    };
}

#coté serveur
NomDuLien {
  pass MotDePasse;
  persist yes;
  device tun0;
  up {
    ifconfig "%% xxx.xxx.xxx.xxx pointopoint xxx.xxx.xxx.xxx";
    route "add -net xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx gw xxx.xxx.xxx.xxx
  };
}
```

Les logs de la connexion se situent dans le répertoire /var/log/daemon.log

De plus, les tunnels Vtun s'ajoutent en tant qu' interfaces réseaux virtuelles vous pouvez donc les afficher grâce à la commande "ifconfig".

```
1:# ifconfig | grep tun
```

Ligne 1: Affichages des interfaces réseaux utilisées pour le tunneling

En cas d'échecs, pour avoir un affichage détaillé de la procédure de connexion ajoutez l'option "-n" à votre commande "client".

## Exécutable et paramètres

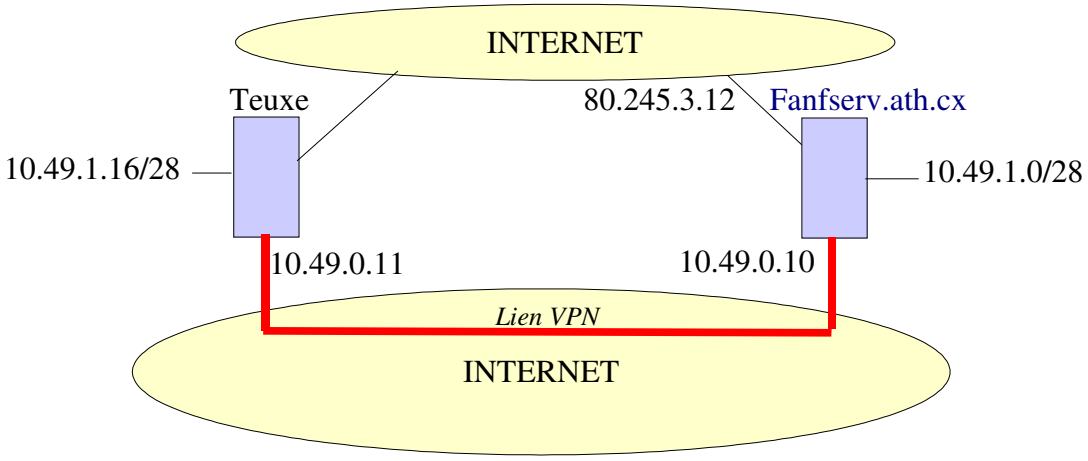
Il n'existe pour vtun qu'un seul exécutable: vtund

*Paramètres principaux:*

| Paramètres | Commentaires |
|---|---|
| vtund -s | Démarrer le serveur |
| vtund NomDuLien xxx.xxx.xxx.xxx | Connecter le lien vpn |
| vtund NomDuLien Nom_Domaine.ath.cx | Connecter le lien vpn |
| Vtund -n NomDuLien xxx.xxx.xxx.xxx | Connecter le lien vpn et affichage de la progression |

# Partie 3: Exemple de configuration

La première mise en place d'un tunnel VPN peux paraître complexe, c'est pourquoi vous trouverez ci-dessous un exemple de configuration d'un tunnel fonctionnel.




Deux réseaux désirent se relier entre eux MaxNet et FredRezo, il a été décidé que Max fera office de serveur pour son réseau, celui de Fred sera donc le client. Quelques détails sur les deux réseaux:

| Nom | IP Internet | Adresse Réseau | IP zone externe | Masque réseau |
|---|---|---|---|---|
| Fanfserv.ath.cx | 80.245.3.12 | 10.49.1.0 | 10.49.0.10 | 255.255.255.240 |
| Teuxe | (non nécéssaire) | 10.49.1.16 | 10.49.0.11 | 255.255.255.240 |

*Fichier de configuration de Fanfserv*

```
option {
   port 5000;                #Port à utiliser
   syslog daemon;            #Fichier de logs à utiliser
       }
default  {
   type tun;         #connection de type tunnel
   proto tcp;        #Utilise UDP
   encr yes;         #Encrypte les donnees
   comp lzo:1;       #Compression LZO de niveau 1
   keepalive yes;  #Reste connecté
   stat yes;         #Statistiques d'utilisations
   speed 0;          #Pas de limite de vitesse, pour limiter mettre valeur en kb 64 / s
        }
fanf-teuxe {
  pass abcdef;      #Mot de passe
  persist yes;      #Connexion persistante (se reconnecte en cas de coupure)
  device tun0;      #Utiliser le périphérique vitruel tun0
  up {
    ifconfig "%% 10.49.0.10 pointopoint 10.49.0.11";
    route "add -net 10.49.1.16 netmask 255.255.255.240 gw 10.49.0.11";
  };
}
```

*Fichier de configuration de Teuxe*

```
option {
   port 5000;     #Port à utiliser
   syslog daemon; #Fichier de logs à utiliser
       }
default  {
   type tun;      #connection de type tunnel
   proto tcp;        #Utilise UDP
   encr yes;         #Encrypte les donnees
   comp lzo:1;     #Compression LZO de niveau 1
   keepalive yes; #Reste connecté
   stat yes;          #Statistiques d'utilisations
   speed 0;        #Pas de limite de vitesse, pour limiter mettre valeur en kb 64 / s
        }
fanf-teuxe {
  pass abcdef;
  persist yes;
  device tun0;
  up {
    ifconfig "%% 10.49.0.11 pointopoint 10.49.0.10";
    route "add -net 10.49.1.0 netmask 255.255.255.240 gw 10.49.0.10";
  };
}
```

Pour que tout fonctionne correctement Max doit lancer le serveur.

```
MaxServ# vtund -s
```

Fred se connectera au serveur créé par Max

```
FredServ# vtund fanf-teuxe fanfserv.ath.cx
```

ou

```
FredServ# vtund fanf-teuxe 80.245.3.12
```

# Chapitre 7: Installation et mise en place de Samba

## A propos de ce chapitre

Dans ce chapitre vous apprendrez comment installer et configurer un serveur Samba pour qu'il fasse office de serveur WINS et de CPD (Contrôleur Principal de Domaine).

*Note: Cette partie du dossier n'est absolument pas indispensable, elle n'est utile que pour afficher les postes dans le voisinage réseau. De plus, étant donné que les failles de sécurité et les virus exploitant les failles de netbios (port 135 notamment) sont très nombreuses, il peut être préférable de bloquer les communications entre les noeuds sur les ports 135 à 139, à vous de décider.*

## Partie 1: Rappels théoriques

Le protocole netbios, permettant d'explorer le voisinage réseau sous Windows et de faire la résolution Nom Machine <-> IP (attention, ce n'est pas un DNS) ne supporte pas le routage, notre réseau étant basé essentiellement sur du routage et des liens vpn, il devient donc impossible pour les utilisateurs windows d'explorer le réseau étendu et d'accéder aux postes grâce à leur nom réseau, ils pourront juste explorer le réseau local.
L'installation d'un serveur WINS (Windows Information Name Service) permettra de résoudre ce problème. Le serveur WINS sous linux est intégré à la partie serveur du logiciel SAMBA.

*Note: SAMBA à été créé à la base pour permettre de réaliser des échanges de fichiers, mais peut aussi faire office de serveur wins et de controleur principal de domaine.*

WINS permet la résolution NomMachine <-> IP, mais ce n'est pas lui qui permet l'affichage des ordinateurs dans le voisinage réseau de windows (ou dans LinNeighborhood sous linux). Le protocole netbios, permettant notamment d'afficher le voisinage réseau, n'est pas routable, c'est-à-dire qu'il ne se propagera pas sur les sous réseaux. Sous un réseau Windows, un système d'élection permet de définir un master navigateur qui sera en quelque sorte le contrôleur du voisinage réseau. Nous allons faire en sorte que nos serveurs linux remportent toujours ces élections et deviennent donc master navigateur ou local navigateur selon le statut du serveur (central ou de noeud). Le réseau étant divisé en plusieurs sous réseaux, nous allons assigner le rôle de master navigateur à notre serveur principal et chaque serveur noeud aura un rôle de local navigateur, chaque local navigateur échangera ses informations avec le master navigateur, de cette manière tout le réseau aura accès aux postes via le voisinage réseau.

*Attention ! : Il doit y avoir au maximum un local navigateur par sous réseau, s'il y en a plus vous risquez de rencontrer des problèmes sur votre réseau.*

## Partie 2: Installation de Samba

```
1:# apt-get install samba
```

Ligne 1: Installation de samba

*Note: il vous sera demandé si vous désirez utiliser Debconf pour configurer samba, répondez non.*
*Note 2: How do you want to run samba ? Daemons*
*Note 3: Create samba password file ? YES*

## Partie 3: Configuration de Samba

La configuration et le partage de répertoires sur le réseau depuis un poste Linux utilisant samba n'entre pas dans les objectifs de ce dossier, vous trouverez plus de détails sur sa configuration sur internet dans les nombreux guides consacrés à ce sujet.

Fichier de configuration:
/etc/samba/smb.conf

*Paramètres communs*

```
[global]
workgroup = AngersWireless        #Nom du goupe de travail ou domaine
name resolve order = wins host bcast        #Ordre de recherche pour la résolution nom<->IP
netbios name = AWServer           #Nom réseau du serveur que vous configurez
server string = Serveur Central   #Commentaire du serveur que vous configurez
invalid users = root              #L'utilisateur root ne peux pas se logguer
security = share                  #Sécurité de type share
guest account = nobody            #Utilisateur à utiliser pour les accès anonymes
encrypt passwords = true          #Utiliser des mots de passe cryptés
os level = 255              #Lors d'une élection passera prioritaire
socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=819
```

*Paramètres serveur : à ajouter si vous configurez un serveur central*

```
local master = yes          #Explorateur maitre local
domain master = yes         #Explorateur maitre de domaine
prefered master = yes       #Lors du démarrage force une élection
wins support = true         #Active le serveur WINS
remote announce = 10.49.1.1/AngersWireless 10.49.1.49/AngersWireless
#faire l'annonce sur tous les serveurs de noeuds
interfaces = 10.49.1.1/16  #Samba fonctionnera sur cette interface
```

*Paramètres serveurs noeuds: à ajouter si vous configurez un serveur de noeud.*

```
Remote announce = 10.49.1.1/AngersWireless #faire l'annonce au serveur central
interfaces = 10.49.1.49/28          #Samba fonctionnera sur cette interface
domain master = no                  #Pas explorateur maitre de domaine
local master = yes                  #Explorateur maitre local
preferred master = yes              #Lors du démarrage force une élection
wins support  = no                  #Ne fait pas serveur WINS
wins server = 10.49.1.1             #Adresse du serveur WINS a utiliser
```

Exécutable serveur:
/etc/init.d/samba
Paramètres de l'exécutable:

| | |
|---|---|
| start | *Démarre le serveur samba.* |
| stop | *Arrête le serveur samba.* |
| reload | *Recharge le fichier de configuration du serveur samba.* |
| Force-reload | *Recharge le fichier de configuration du serveur samba sans prendre en compte les erreurs.* |
| restart | *Redémarre le serveur samba* |

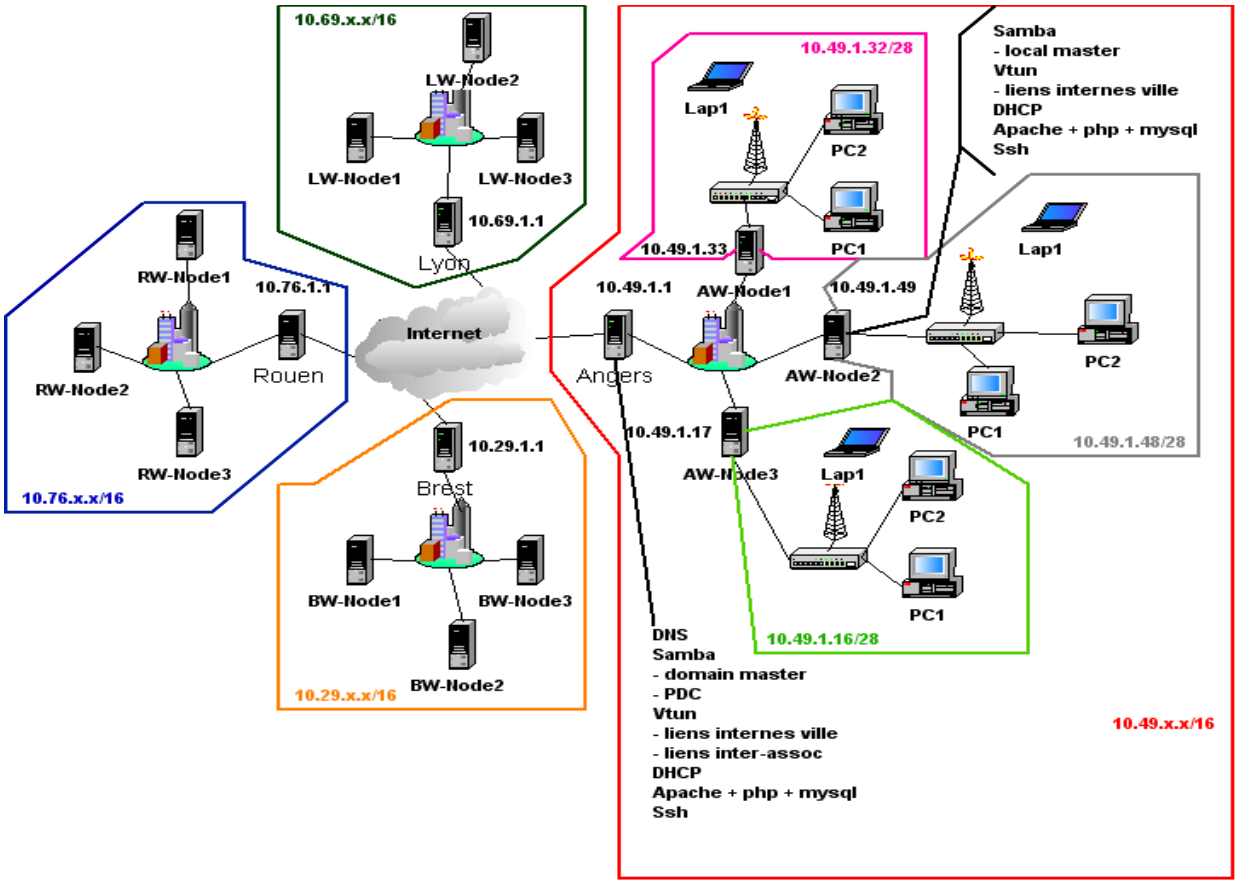
Exemple d'utilisation, démarrage du serveur:

```
1:# /etc/init.d/samba start
```

*Ligne 1: Démarrage du serveur samba*

A ce point du dossier, nous possédons un serveur  et un réseau fonctionnel mais non sécurisé, en effet n'importe qui peut s'y connecter.

# Chapitre 8: Mise au point




Nous allons maintenant identifier les services proposés par notre réseau au point à ce niveau de la documentation.
Différents services sont maintenant opérationnels sur les serveurs.
Le serveur central possède des services à destination du réseau global (DNS, Samba, Apache, Vtun, Wins, PDC) et de son réseau local (DHCP, Ssh).
Les nodes possèdent eux aussi quelques services (Samba, DHCP, Apache, Vtun).
Tous les noeuds sont reliés entre eux, un poste situé sur un noeud peut communiquer avec un autre poste situé sur un autre noeud.
Un poste situé sur un noeud pourra, sous Windows, grâce à son voisinage réseau afficher
tous les postes connectés au réseau étendu.

# Chapitre 9: Installation et mise en place de Zebra, Ospf et bgp

## A propos de ce chapitre

Dans ce chapitre vous découvrirez quelques notions et explications sur le routage ainsi qu'une méthode pour installer une solution de routage dynamique.

## Partie 1: rappels sur le routage

Le routage permet d'indiquer aux requêtes ip par où passer pour aller à une certaine destination. Chaque ordinateur possède sa propre table de routage, qui peux être très simple dans le cas d'un client ou complexe dans le cas d'un serveur ou d'un routeur.
Une route dans une table de routage contient 4 paramètres principaux:

- L'adresse du réseau de destination.
- Son masque réseau.
- L'adresse de la passerelle permettant d'accéder à ce réseau.
- L'interface communiquant avec la passerelle.

Voici un exemple de table de routage

Table de routage IP du noyau

| | Destination | Passerelle | Genmask | Indic | Metric | Ref | Use | Iface |
|---|---|---|---|---|---|---|---|---|
| *1:* | 10.44.0.102 | * | 255.255.255.255 | UH | 0 | 0 | 0 | tun0 |
| *2:* | 10.49.1.0 | * | 255.255.255.240 | U | 0 | 0 | 0 | eth0 |
| *3:* | localnet | * | 255.255.255.0 | U | 0 | 0 | 0 | eth0 |
| *4:* | 10.44.0.0 | 10.44.0.102 | 255.255.0.0 | UG | 0 | 0 | 0 | tun0 |
| *5:* | default | 192.168.1.254 | 0.0.0.0 | UG | 0 | 0 | 0 | eth0 |

*Ligne 1: Les requêtes à destination de l'ip 10.44.0.102 seront redirigées sur l'interface tun0*
*Ligne 2: Les requêtes à destination du réseau 10.49.1.0/28 seront redirigées sur l'interface eth0.*
*Ligne 3: Les requêtes à destination du réseau local seront redirigées sur l'interface eth0*
*Ligne 4: Les requêtes à destination du réseau 10.44.0.0/16 seront redirigées sur la passerelle ayant pour adresse IP 10.44.0.102 par l'intermédiaire de l'interface tun0*
*Ligne 5: Toutes les requêtes ne pouvant êtres résolues par une des lignes de la table de routage seront renvoyées sur la passerelle ayant pour adresse IP 192.168.1.254 par l'intermédiaire de l'interface eth0*

La route peut être obtenue sous windows en tapant la commande "route print" ou sous linux en tapant la commande "route".

Mais ce système de routage reste très limité pour les grands réseaux et tout particulièrement les réseaux étendus et composés de multiples sous-réseaux.
Les 2 principaux protocoles utilisés sur Internet sont l'OSPF (Définit dans la RFC 2328) et le standard BGP4 (Définit par la RFC 1771).
Le protocole OSPF sera utilisé pour l'échange de tables entre les différents nodes, et entre les nodes et le serveur d'une communauté, le protocole BGP sera quant à lui utilisé pour l'échange des tables de routage entre les communautés.

Le protocole OSPF (Open Shortest Path First) a été développé pour faciliter le routage sur de très grands réseaux, notamment Internet.
Ce protocole permet, mot à mot, de trouver le chemin le plus court vers une destination sur un réseau ouvert; le tout grâce à du routage dynamique.

Les 2 principales fonctionnalités: permettre de trouver le chemin le plus court à travers toute une série de routeurs et publier automatiquement les tables de routages, pour, par exemple, mettre à jour le routage lors de l'ajout de routeurs sur le réseau. Si nous appliquons ce principe à notre projet, nous pouvons nous rendre compte que l'utilité de la fonctionnalité permettant de trouver le chemin le plus court entre deux points sera le plus souvent inutile; en effet tous les points étant reliés entre eux les chemin le plus court entre 2 noeuds du réseau sera forcément le lien direct, sauf en cas de rupture d'un lien direct. Par contre la fonctionnalité permettant la mise à jour automatique est très intéressante pour tout ce qui concerne l'ajout de lien entre les communautés, dès qu'un noeud possède une nouvelle route vers une autre communauté, les informations de routage de tous les autres noeuds seront automatiquement mises à jour et les autres noeuds seront en mesure de commiquer automatiquement avec la nouvelle communauté par l'intermédiaire du noeud ayant rajouté le lien.

Le BGP4 (Border Gateway Protocol Version 4) définit en 1995 par la RFC 1771 est un protocole de routage dynamique, et permet la distribution de tables de routage entre des routeurs BGP4. Chaque routeur BGP4 possède un AS (Autonomous System number). Les AS sont utilisés sur internet et la plage 1 -> 64511 est réservé aux routeurs d'internet. Une plage publique a été définie, de 64512 -> 65535. Nous allons utiliser la plage 651xx (65100 + numéro de département).
Les AS peuvent être comparés à des identifiant de zone, chaque routeur s'échangeant les routes lui permettant de communiquer avec d'autres zones.

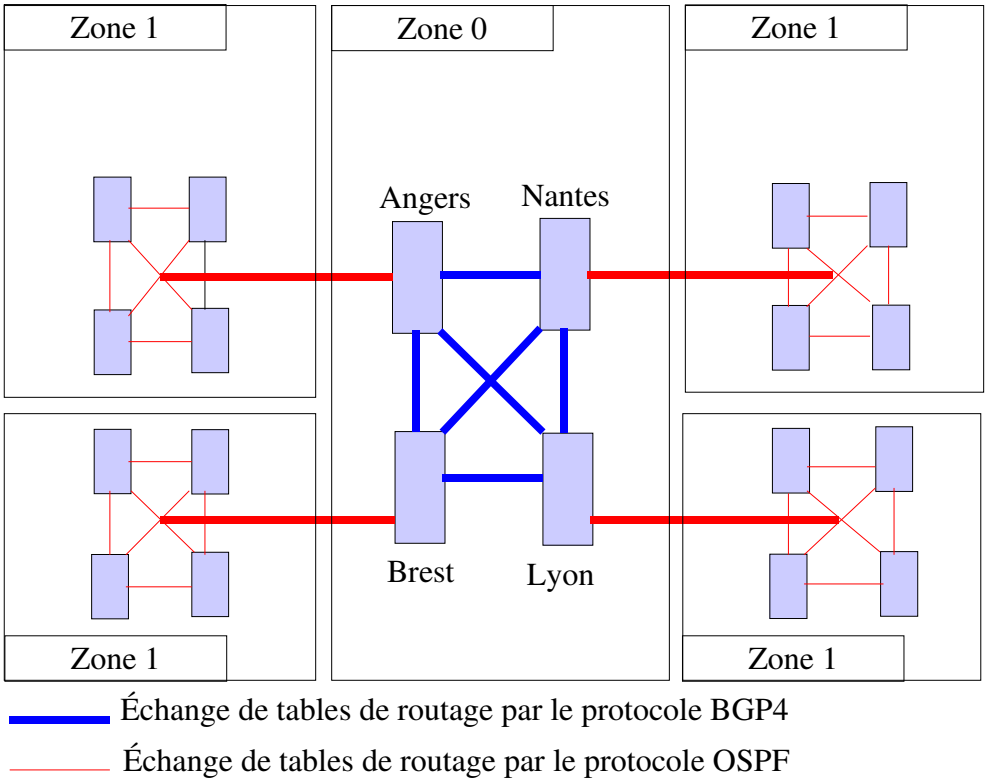
Pour mettre en oeuvre l'OSPF et le BGP4, il existe un logiciel fonctionnant sous Linux appelé Zebra (www.zebra.org)

## Partie 2: Fonctionnement de Zebra

Zebra fonctionne grâce à un système de hiérarchie. Les réseaux sont reliés ensemble par une zone appelée "épine dorsale" ou "zone 0 ". Tout le trafic passe par la zone zéro, et tous les routeurs de cette zone posséderont les informations de routage de toutes les autres zones. Dans notre cas la zone 0 sera composée de tous les serveurs de toutes les associations départementales. (10.44.1.1 pour Nantes, 10.49.1.1 pour Angers, etc ...). De plus, comparé à RIP .(Routing Internet Protocole) les informations sont propagées très rapidement.

Dans un souci d'économie de bande passante, le méthode d'envoi de données (de tables de routages) se fait par la méthode de multi-distribution et non de diffusion, seuls les postes nécessitant des informations de routage en recevront. De plus les routeurs ne recevront que des tables de routage les concernant, grâce au fonctionnement par zone.
Les routeurs ayant des interfaces dans 2 zones différentes sont appelés "Area Border Routers".
Angers possédera un Area Border Router, en effet le serveur 10.49.1.1 sera en

communication avec le réseau 10.49.X.X (la zone 1) et avec le réseau 10.X.X.X (la zone 0).




*Chapitre 9 – Partie 2 – Schéma 1: Fonctionnement du routage*

Pour chaque protocole, lors de la configuration, l'utilisateur choisit quelles seront les tables de routage qui seront redistribuées .
Pour le protocole OSPF, lors de la configuration sur un des serveurs de la zone 0, nous indiquerons à ospfd de redistribuer les routes BGP, Statiques et OSPF vers les noeuds de la zone 1. Par conséquent les noeuds de zone 1 sauront comment accéder aux réseaux qui sont derrière des serveurs de zone 0.
Par contre pour le protocole BGP4, lors de la configuration d'un serveur de la zone 0, nous n'indiquerons pas de redistribuer les routes OSPF. En effet, il est inutile de redistribuer au serveurs de zone 0 (et donc automatiquement aux noeuds de zone 1), les routes permettant d'atteindre un sous réseau précis d'un noeud. Une route globale vers le serveur central d'une communauté est suffisante.

## Partie 3: Installation de Zebra et Ospf

*Note: Attention, les différentes versions de Zebra ne sont pas compatible entre elles, tout le réseau doit donc utiliser la même version de zebra*

La version utilisée par la communauté wireless francophone au moment où ce document est écrit est la 0.94, elle est disponible en téléchargement sur le site officiel de zebra (www.zebra.org).

```
1:# wget ftp://ftp.zebra.org/pub/zebra/zebra-0.94.tar.gz
2:# tar xfvz zebra-0.94.tar.gz
3:# cd zebra-0.94
4:# ./configure --sysconfdir=/etc/zebra
5:# make; make install
```

Ligne 1: Téléchargement de l'archive du logiciel
Ligne 2: Décompression de l'arcihve du logiciel
Ligne 3: Déplacement dans le repertoire
Ligne 4: Configuration pré-compilation, on indique ici le répertoire où seront situés les fichiers de configuration
Ligne 5: Compilation et installation de zebra

Nous allons maintenant configurer zebra pour l'adapter à la configuration du réseau.

## Partie 4: Configuration de Zebra et Ospf

### Configuration du serveur central

Fichier /etc/zebra/zebra.conf

```
hostname awserver
password xxxxxx
enable password azerty
log file /var/log/zebra/zebra.log
log stdout
!
interface lo
!
interface eth0
!
interface eth0:0
!
interface tun0
!
interface tun1
!
interface tun2
!
interface tun3
!
interface tun4
!
interface tun5
!
interface tun6
!
interface tun7
!
interface tun8
!
line vty
!
```

Nom du serveur
Mot de passe pour la connexion telnet
Mot de passe de type azerty
Fichier de log
Afficher à l'écran les actions exécutées par zebra
Début de la déclaration des interfaces
Déclarez ici toutes vos interfaces de tunnels.

*Note: Les commentaires ne sont pas pris en compte dans les fichiers de configuration de zebra, ospf et bgp, ne faites donc pas précéder vos lignes d'un "#".*

Ce fichier de configuration est relativement simple, détaillez-y vos interfaces physiques, vos alias et vos tunnels .

*Note: Tapez "ifconfig" dans un shell pour obtenir la liste de vos interfaces.*

Fichier /etc/zebra/ospfd.conf

```
hostname awserver                                          Nom du serveur sur lequel est lancé le serveur
password xxxxxx                                            Mot de passe pour la connexion telnet
enable password azerty                                     Mot de passe de type azerty
debug ospf packet all                                      Mode debug (affichage à l'écran de ce qui se passe)
!
interface lo                                               Début de déclaration des interfaces
!
interface eth0
!
interface eth0:0
!
interface tun1                                             Ne déclarez que les tunnels interne à votre communauté, il ne faut pas
        description Tunnel -> Lessyv                       déclarer dans le fichier ospfd.conf les interfaces tunnel vers les autres
        ip ospf network point-to-point                     communautés, en effet le protocole OSPF ne sera ici utilisé que pour
!                                                          l'échange de routes à l'intérieur de la communauté.
interface tun2
        description Tunnel -> Onyme
        ip ospf network point-to-point
!
interface tun3                                             La description n'est la que pour faciliter le traitement des fichiers de logs
        description Tunnel -> Fanfoue                      Utilisation du protocole OSPF
        ip ospf network point-to-point
!
interface tun4
        description Tunnel -> Gwe
        ip ospf network point-to-point
!
interface tun5
        description Tunnel -> Quinq
        ip ospf network point-to-point
!
interface tun6
        description Tunnel -> Lolo
        ip ospf network point-to-point
!                                                          Fin de la déclaration des interfaces
router ospf                                                Configuration du routeur de type OSPF
        ospf router-id 10.49.1.1                           Adresse IP du routeur ospf
        passive-interface eth0                             Interfaces physiques du serveur
        passive-interface eth0:0
        network 10.49.0.14/32 area 65149                   L'area 65149 sera diffusée via l'interface de tunnel 10.49.0.14
        network 10.49.0.17/32 area 65149
        network 10.49.0.19/32 area 65149
        network 10.49.0.23/32 area 65149
        network 10.49.0.25/32 area 65149
        network 10.49.0.27/32 area 65149
        redistribute static                                Les routes statiques seront redistribuées par le protocole OSPF
        redistribute connected                             Les routes connectés seront redistribuées par le protocole OSPF
        redistribute bgp                                   Les routes bgp seront redistribuées par le protocole OSPF
        distribute-list OUT_CONNECTED out connected        Création d'une liste de distribution pour les routes connectées
        distribute-list OUT_STATIC out static              Création d'une liste de distribution pour les routes statiques
        distribute-list OUT_BGP out bgp                    Création d'une liste de distribution pour les routes bgp
        !
        access-list OUT_CONNECTED permit 10.49.0.0/16      Redistribuer les routes connectées provenant du réseau 10.49.0.0/16
        access-list OUT_CONNECTED deny any                 Interdire les autres
        !
        access-list OUT_STATIC permit 10.0.0.0/8           Redistribuer les routes statiques provenant du réseau 10.0.0.0/8
        access-list OUT_STATIC deny any                    Interdire les autres
        !
        access-list OUT_BGP permit 10.0.0.0/8              Redistribuer les routes provenant du réseau 10.0.0.0/8
        access-list OUT_BGP deny any                       Interdire les autres
!
line vty
!
```

Note: Pour remplir ce fichier aidez vous de votre /etc/vtund.conf

Comme vous pouvez le constater la configuration du fichier ospfd.conf est relativement simple.

Fichier /etc/zebra/bgpd.conf

| | |
|---|---|
| *hostname awserver* | Nom du serveur |
| *password **xxxxxx*** | Mot de passe pour la connexion telnet |
| *enable password azerty* | Mot de passe de type azerty |
| *log stdout* | Afficher à l'écran les actions exécutées par zebra |
| *log file /var/log/zebra/bgpd.log* | Fichier de log |
| *!* | |
| *debug bgp events*<br>*debug bgp filters*<br>*debug bgp fsm*<br>*debug bgp keepalives*<br>*debug bgp updates* | Début de déclaration des actions à afficher (fichiers de log et écran) |
| *!* | |
| *router bgp 65149* | Configuration du routeur de type bgp |
| *!* | |
| *bgp router-id 10.49.1.1* | Adresse IP du routeur |
| *!* | |
| *network 10.49.0.0/16* | Réseau sur lequel fonctionne le routeur |
| *!* | |
| *redistribute static* | Redistribuer par le protocole BGP les routes statiques |
| *!* | |
| *neighbor 10.44.0.102 remote-as 65144*<br>*neighbor 10.44.0.102 distribute-list local_nets in* | Envoyer les informations de routage à une autre communauté ayant pour IP 10.44.0.102 et pour AS 65144 |
| *!* | |
| *access-list local_nets permit 10.0.0.0/8*<br>*access-list local_nets deny any* | Ne seront redistribuées que les routes provenant du réseau 10.0.0.0/8, interdire les autres. |
| *!* | |

## Configuration d'un noeud

Pour remplir ces fichiers, les commentaires sont les mêmes que pour les fichiers du serveur.

Fichier /etc/zebra/zebra.conf

```
hostname gweserv
password  xxxxxxx
enable password azerty
#log file /var/log/zebra/zebra.log
log stdout          #Affiche les informations à l'écran (utile pour les tests)
!
interface lo
!
interface eth0
!
interface tun0
!
interface tun1
!
interface tun2
!
line vty
!
```

Fichier /etc/zebra/ospfd.conf

```
hostname awserver
password xxxxxxx
enable password azerty
debug ospf packet all
log file /var/log/zebra/ospfd.log
log stdout
!
interface lo
!
interface eth0
!
interface tun0
        description Tunnel -> Serveur
        ip ospf network point-to-point
!
interface tun1
        description Tunnel -> Lolo
        ip ospf network point-to-point
!
interface tun2
        description Tunnel -> Onyme
        ip ospf network point-to-point
!
router ospf
      ospf router-id 10.49.1.81
      passive-interface eth0
      network 10.49.0.22/32 area 65149
      network 10.49.0.28/32 area 65149
      network 10.49.0.32/32 area 65149
      redistribute static
      redistribute connected
      distribute-list output_connected out connected
      distribute-list output_static out static
      area 49 import-list filtre_local
!
access-list output_connected permit 10.49.0.0/16
access-list output_connected deny any
!
access-list output_static permit 10.0.0.0/8
access-list output_static deny any
!
access-list filtre_local permit 10.0.0.0/8
access-list filtre_local deny any
!
line vty
!
```

## Partie 5: Lancement des logiciels

Nous allons maintenant lancer et tester les logiciels.

### Lancement sur le serveur central

```
1:# zebra -d &
2:# ospfd -d &
3:# bgpd -d &
```

Ligne 1: Lancement du logiciel zebra
Ligne 2: Lancement du logiciel ospfd
Ligne 3: Lancement du logiciel bgpd

Note: Il est important de respecter l'ordre, lancer zébra en premier puis ospfd et bgpd

```
1:# telnet awserver 2601
Trying 127.0.0.1...
Connected to awserver.
Escape character is '^]'.
Hello, this is zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.
User Access Verification
Password:
awserver>
2:# awserver> show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route
O>* 10.49.0.26/32 [110/20] via 10.49.0.27, tun6, 00:11:35
O   10.49.0.27/32 [110/10] is directly connected, tun6, 00:19:36
O   10.49.1.112/28 [110/20] via 10.49.0.27, tun6, 00:11:34
3:# awserver> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route
B   10.29.0.0/16 [20/0] via 10.29.1.1, tun101, 00:11:52
B>* 10.31.1.0/24 [20/0] via 10.44.0.102, tun100, 00:20:16
B   10.44.0.0/16 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.69.0.0/16 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.69.2.0/24 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.69.20.0/24 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.69.25.0/24 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.69.254.16/28 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.72.1.0/24 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.91.0.0/16 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.1/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.6/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.13/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.18/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.26/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.32/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.104/32 [20/0] via 10.44.0.102, tun100, 00:20:16
B>* 10.252.69.105/32 [20/0] via 10.44.0.102, tun100, 00:20:16
4:# awserver> exit
```

Ligne 1: Connexion en telnet sur le logiciel zébra du serveur.
Ligne 2: Permet d'afficher les routes obtenues grâce au protocole ospf
Ligne 3: Permet d'afficher toutes les routes obtenues grâce au protocole bgp

Nous pouvons nous rendre compte que la majorité des routes ont pour gateway l'adresse IP 10.44.0.102 qui correspond à l'adresse de tunnel du

serveur de la communauté de Nantes. Les routes ont donc bien étés correctement automatiquement diffusées.
Plusieurs autres outils sont disponibles, principalement en telnet pour consulter l'état des routes notamment:

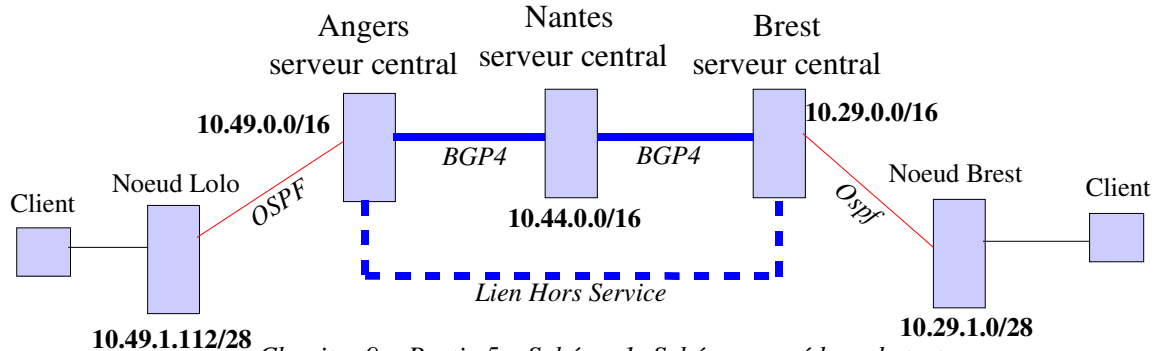
Telnet awserver 2603 vous permettra de vous connecter à l'interface pour le protocole OSPF
Telnet awserver 2605 vous permettra de vous connecter à l'interface pour le protocole BGP4

### Lancement sur un noeud

Utilisez les même commandes pour lancer zebra et ospf sur un noeud.
Vous pouvez aussi vérifier si vous avez bien reçu les routes grâce à la commande "show ip route ospf" en telnet.

### Test de la solution

Nous allons maintenant vérifier le bon fonctionnement du routage et réaliser quelques scans réseau grâce au logiciel Nmap (apt-get install nmap).
Nous allons réaliser ces tests sur un réseau distant qui n'est pas "physiquement" relié à notre réseau. Nous passerons donc par le réseau d'une autre communauté qui servira en quelque sorte d'intermédiaire. Tout ceci pour montrer ce qui pourra se passer en cas de panne d'un des serveurs, ou d'un des tunnel vpn.
Ci-dessous vous trouverez un schéma détaillant les conditions de test.




*Chapitre 9 – Partie 5 – Schéma 1: Schéma procédure de tests*

Depuis le noeud Lolo, nous allons faire un scan du réseau pour avoir une idée des adresses IP des postes connectés sur le réseau de Brest (seulement sur la plage 10.29.1.0/24 pour un gain de temps)

```
1#: LoLoServ:~# nmap -sP 10.29.1.0/24
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host  (10.29.1.1) appears to be up.
Host  (10.29.1.3) appears to be up.
Host  (10.29.1.4) appears to be up.
Host  (10.29.1.25) appears to be up.
Host  (10.29.1.100) appears to be up.
Host  (10.29.1.101) appears to be up.
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 104 seconds
```

Ligne 1: Lancement d'un scan de type ping sur le réseau 10.69.1.0/24

Comme nous pouvons le voir 6 postes ont étés découverts, nous allons maintenant déterminer par ou sont passés les paquets pour aller à cette destination.

1#: **Client Lolo sous windows:~# tracert 10.29.1.4**

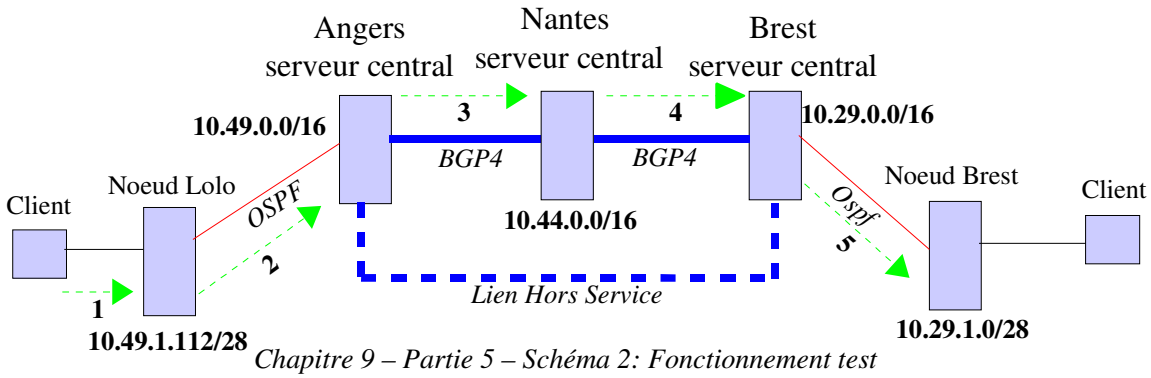
*Détermination de l'itinéraire vers 10.69.1.4 avec un maximum de 30 sauts.*

| | | | | |
|---|---|---|---|---|
| *1* | *<1 ms* | *<1 ms* | *<1 ms* | *LOLOSERVER [10.49.1.81]* |
| *2* | *186 ms* | *221 ms* | *152 ms* | *AWSERVER [10.49.0.22]* |
| *3* | *234 ms* | *141 ms* | *172 ms* | *NWSERVER [10.44.1.17]* |
| *4* | *444 ms* | *323 ms* | *227 ms* | *BWSERVER [10.29.1.1]* |
| *5* | *890 ms* | *644 ms* | *432 ms* | *10.29.1.4* |

*Itinéraire déterminé.*

Ligne 1: Lancement de tracert depuis un client du réseau utilisant le système d'exploitation windows

Nous pouvons bien voir ici l'itinéraire de la requête, elle passe tout d'abord par le noeud auquel elle est reliée, puis du noeud passe au serveur central d'Angers, puis le serveur central de Nantes, ensuite le serveur central de Brest et enfin le noeud sur lequel on fait le tracert.




*Chapitre 9 – Partie 5 – Schéma 2: Fonctionnement test*

## Partie 6: Reconfiguration des liens vtun

Grâce à BGP4 le routage entre les communautés s'effectue automatiquement, pour éviter les conflits de routes, il est donc indispensable de supprimer les lignes du fichier /etc/vtund.conf créant des routes statiques entre les communautés.
Exemple pour la route avec Nantes:

*Fichier /etc/vtund.conf*

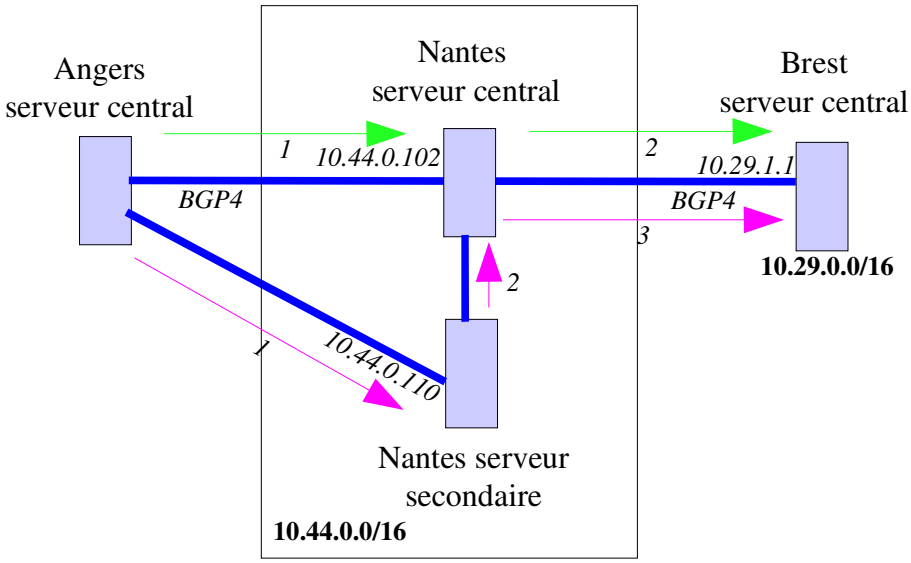
```
AW-NW {
  pass xxxxxx;
  persist yes;
  device tun100;
  up {
    ifconfig "%% 10.49.0.101 pointopoint 10.44.0.102";
#     route "add -net 10.44.0.0 netmask 255.255.0.0 gw 10.44.0.102";
  };
 }
```

Ici la route est seulement commentée, mais vous pouvez la supprimer.

Note: Pensez bien à relancer vtund avec l'option -s et à relancer les liens actifs pour que les modifications puissent êtres prises en compte.

## Partie 7: Simulation de panne pendant une activité réseau

Nous allons dans cette partie simuler la panne d'un lien pendant un transfert de données (représentée ici par un ping) entre plusieurs noeuds distants.




*Chapitre 9 – Partie 7 – Schéma 1: Routage dynamique*

Le lien entre Angers et Brest est Hors-Service, ainsi que les liens entre le serveur secondaire de Nantes et les serveurs de Brest, nous allons maintenant simuler en direct la panne entre le serveur principal d'Angers et de Nantes.

Avant la panne: →

```
FanfServ:~# traceroute 10.29.1.1
traceroute to 10.29.1.1 (10.29.1.1), 30 hops max, 38 byte packets
 1  10.44.0.102 (10.44.0.102)  340.383 ms  186.845 ms  199.158 ms
 2  10.29.1.1 (10.29.1.1)  749.979 ms  821.975 ms  591.962 ms
```

*Ligne 1: Lancement d'un traceroute vers Brest*

Affichage de la panne.

```
FanfServ:~# ping -R 10.29.1.1

64 bytes from 10.29.1.1: icmp_seq=0 ttl=63 time=994.2 ms
RR:  10.49.1.1          #Itinéraire allée
     10.44.0.102
     10.29.1.1
     10.29.1.1          #Iténéraire retour
     10.44.0.102
     10.49.1.1
64 bytes from 10.29.1.1: icmp_seq=1 ttl=63 time=1065.4 ms     (same route)
64 bytes from 10.29.1.1: icmp_seq=2 ttl=63 time=998.5 ms      (same route)
64 bytes from 10.29.1.1: icmp_seq=3 ttl=63 time=1142.2 ms     (same route)          #La coupure se produit ici,  regardez les
                                                                                    numéros d'icmp_seq, ils passent de 3 à 56
64 bytes from 10.69.1.1: icmp_seq=56 ttl=62 time=1108.1 ms
RR:    10.49.1.1        #Itinéraire allée
     10.44.0.110
     10.44.0.102
     10.29.1.1
     10.29.1.1          #Itinéraire retour
     10.44.0.102
     10.44.0.110
     10.49.1.1
64 bytes from 10.29.1.1: icmp_seq=57 ttl=62 time=1127.9 ms    (same route)
64 bytes from 10.29.1.1: icmp_seq=58 ttl=62 time=770.0 ms     (same route)
64 bytes from 10.29.1.1: icmp_seq=59 ttl=62 time=563.5 ms     (same route)
....
--- 10.29.1.1 ping statistics ---
63 packets transmitted, 14 packets received, 77% packet loss
round-trip min/avg/max = 551.9/912.6/1142.2 ms
```

Nous pouvons voir sur cette requête ping la route empruntée par les paquets en allant et en revenant de la destination. Il y a comme vous pouvez le constater énormément de pertes, correspondant aux requêtes ping perdues lors de l'actualisation des routes, en effet il y a un petit délai (environ 20s) avant que zebra ne détecte qu'un lien est inactif et que les routes soient mises à jour. Ce délai est très important, sur internet les échanges de routes entre routeurs et leur réactualisation est quasi instantanée et invisible pour les utilisateurs, en effet les débits sont ici d'environ 128kb/s et sur internet ils sont de plusieurs Gb/s.

Après la panne: →

```
FanfServ:~# traceroute 10.29.1.1
traceroute to 10.29.1.1 (10.29.1.1), 30 hops max, 38 byte packets
 1  10.44.0.110 (10.44.0.110)  258.392 ms  229.741 ms  379.645 ms
 2  10.44.0.102 (10.44.0.102)  322.908 ms  337.908 ms  520.755 ms
 3  10.29.1.1 (10.29.1.1)  1101.955 ms  1175.990 ms  1334.637 ms
```

*Ligne 1: Lancement d'un traceroute vers Brest*

Comme vous pouvez le constater zebra et bgp ont bien actualisé les informations et la communication est maintenant possible mais avec un intermédiaire de plus.

# Chapitre 10: Installation et mise en place de Netfilter (iptables)

## A propos de ce chapitre

Dans ce chapitre vous apprendrez comment fonctionne Netfilter (iptables) et vous apprendre à créer un script permettant de le configurer.
Netfilter (iptables) est un logiciel de firewalling permettant de sécuriser un serveur linux.

Nous allons sécuriser au maximum le serveur, en interdisant tout accès non prévu, que ce soit depuis l'extérieur du réseau (Internet) vers l'intérieur ou de l'intérieur du réseau vers Internet.
Il est indispensable de créer un script qui sera exécuté à chaque démarrage de la machine, en effet un ensemble de règles iptables est généralement très important.

## Partie 1: Fonctionnement d'iptables

Iptables est principalement composé de 3 tables: NAT, FILTER et MANGLE
Chacune de ces tables ont des commandes (chaines) et des usages spécifiques.

### Les Tables

| Table | Définition | Chaines | Définition Chaine | Cibles | Définition cibles |
|---|---|---|---|---|---|
| NAT | *Table utilisée pour la translation d'adresse ou la translation de ports* | PREROUTING | *Permet de spécifier "à l'arrivée du firewall"* | DNAT | *Indique l'adresse de destination du paquet* |
| | | POSTROUTING | *Permet de spécifier "à la sortie du firewall"* | SNAT | *Indique l'adresse source du paquet* |
| | | | | MASQUERADE | *Ce paramètre permet de faire croire aux sites visités (par exemple) que les requêtes proviennent de lq passerelles qlors qu'ils proviennent de son réseau local.* |
| | | | | | |
| FILTER | *Table utilisée par défaut si rien n'est spécifié. Contient toutes les règles de filtrage , chaque règle contient un cible indiquant comment traiter le paquet.* | FORWARD | *Pour les paquets passant par la passerelle (concerne la table NAT principalement)* | ACCEPT | *Accepte les paquets* |
| | *Il faut utiliser cette table pour tout ce qui concerne le routage,* | INPUT | *Pour les paquets entrant dans la passerelle* | DENY | *Refuse les paquets* |
| | | OUTPUT | *Pour les paquets sortant de la passerelle* | DROP | *Refuse un paquet mais ne renvoie pas d'erreur à l'utilisateur.* |
| | | | | REJECT | *Refuse un paquet et renvoie une erreur à l'utilisateur* |
| | | | | | |
| MANGLE | *Table contenant les règles concernant la modification de paquets* | | | | |
| | | | | | |

## Les Paramètres de la commande iptables

Il n'existe pas d'interface "officielle" permettant de traiter les règles iptables, tout se fait à partir de commandes plus ou moins complexes que nous allons détailler dans cette partie.

| Commande | alternative | Commentaires |
|---|---|---|
| -A | --append | Ajoute la règle à la fin de la chaine spécifiée (INPUT, PREROUTING, FORWARD, ...) |
| -D | --delete | Supprime la chaine spécifiée, fonctionne soit avec le numéro de chaine soit avec la chaine complète |
| -R | --replace | Idem à -D mais pour remplacer la chaine |
| -I | --insert | |
| -F | --flush | Permet de vider toutes les tables, possibiliter de ne vider que par chaine |
| -N | --new-chain | Permet de créer une nouvelle chaine |
| -X | --delete-chain | Permet d'effacer une chaine |
| -P | --policy | Permet de définir la cible par défaut d'une chaine (INPUT, PREROUTING, FORWARD, ...) |
| -p | --protocol | Permet de spécifier un protocole (tcp, udp, icmp,all) |
| -s | --source | Spécifie une adresse source |
| -d | --destination | Spécifie une adresse de destination |
| -i | --in-interface | Spécifie une interface d'entrée |
| -o | --out-interface | Spécifie une interface de sortie |
| -f | --fragment | Paquet fragmenté |
| -sport | --source-port | Spécifier le port source ou une liste de ports source, syntaxe: xxxx:xxxx |
| -m multiport | | Ajouté à -sport ou -dport permet de spécifier plusieurs port, syntaxe: xxx,xxx,xxx |
| -dport | --destination-port | Spécifier le port de destination ou une plage de ports |
| --tcp-flags | | Spécifier un flag tcp à matcher (SYN, ACK, FIN, RST, URG, PSH, ALL, NONE) |
| --icmp-type | | Spécifier un type de paquet icmp à traiter |
| --mac-source | | Spécifier une adresse MAC à traiter |
| --state | | Permet de spécifier l'état du parque à traiter parmi les états suivants:<br>- ESTABLISHED: paquet associé à une conexion déjà établie<br>- NEW: paquet demandant une nouvelle connexion<br>- INVALID: paquet associé à une connexion inconnue<br>- RELATED: nouvelle connexion mais lié à autre chose |
| --to-destination | *NAT seulement* | Spécifie une adresse de destination pour une translation NAT |
| --log-level | *Pour les LOGS* | Niveau de Log |
| --log-prefix | | Permet de spécifier un préfix pour les logs |

*Note: tous les paramètres décrits dans le tableau ci dessus ne seront pas obligatoirement utiles pour notre projet.*

## Partie 2: Création du script iptables

Nous allons maintenant étudier nos besoins en règles iptables, nous prendrons un cas d'utilisation et nous transformerons chacune des phrases de la problématique en règle iptables.

Nous allons commencer par définir tous les différents éléments du réseau pour pouvoir leur appliquer des règles spécifiques.
Pour chaque élément du réseau nous allons créer une variable, ce qui permettra de n'avoir à changer que le contenu des variables et non tout le script.

Nous allons maintenant rechercher les ports utilisés par tous les services que nous avons utilisé pour notre projet.

| **SERVICE** | **PORT (s)** | | **SERVICE** | **PORT (s)** | | **SERVICE** | **PORT (s)** |
|---|---|---|---|---|---|---|---|
| SAMBA | 135-139 | | MYSQL | 3306 | | IM (mess instantanée) | 5222 . |
| DNS | 53 | | RADIUS | 1812 . 1813 | | Jeux (q3, HL, ...) | 27000-29000 |
| WINS | | | ZEBRA | 2601 | | | |
| MAIL | 110.25. 119.143 . 995 | | OSPF | 2603 | | | |
| SSH | 22 | | BGP4 | 2605 | | | |
| FTP | 21 | | VTUN | 5000 | | | |
| HTTP | 80 .443 | | PPP | | | | |
| SNMP | 161 | | PPTP | | | | |
| TELNET | 23 | | IRC | 6660-6667 | | | |

```
#! /bin/sh
#on commence par définir les différens réseaux et Ips des serveurs
IP_SERVEUR_PRIVE = 192.168.1.50/32 # à changer en fonction du réseau
RESEAU_GLOBAL_AW = 10.49.0.0/16
RESEAU_GLOBAL = 10.0.0.0/8
RESEAU_NODE = 10.49.1.0/28 # à changer en fonction du noeud
IP_NODE = 10.49.1.1/32 # à changer en fonction du noeud

#Puis on définit les ports autorisés vers le réseau global de la communauté
OKPORTS_GLOBAL_TCP =
OKPORTS_GLOBAL_UDP =
#Ports autorisés vers le réseau de la communauté
OKPORTS_AW_TCP =  80,22,21,2601, 135-139,
OKPORTS_AW_UDP =
#Ports autorisés vers le net
OKPORTS_NET_TCP = 80,22
OKPORTS_NET_UDP =
#Ports autorisés en INPUT / OUTPUT sur le node
NODE_PORTS_TCP =
NODE_PORTS_UDP =
#Interface reliée au net
IFNET = eth0
#Interface reliée au réseau communautaire
IFAW = eth0:0
#Interface du NAS
IFNAS = eth1

# on vide toutes les tables existantes
iptables -F

# on définit une politique DROP donc restrictive par défaut
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#On active le nat en sortie sur l'interface eth0
iptables -t nat -A POSTROUTING -o eth0

#On définit les règles qui seront appliquées en TCP et UDP du réseau local vers le net
iptables -A FORWARD -s $RESEAU_LOCAL -j ACCEPT

#On définit les règles qui seront appliquées à destination d'internet pour les ports du noeud
iptables -A FORWARD -s $RESEAU_NODE -p tcp -m multiport –dports $OKPORTS_NET_TCP -o IFNET -j ACCEPT
iptables -A FORWARD -s $RESEAU_NODE -p udp -m multiport –dports $OKPORTS_NET_UDP -o IFNET -j ACCEPT

#On définit les règles qui seront appliquées en TCP et UDP depuis le réseau du node vers le réseau global
iptables -A FORWARD -s $RESEAU_NODE -p tcp -m multiport –dports $OKPORTS_GLOBAL_TCP -d $RESEAU_GLOBAL -j ACCEPT
iptables -A FORWARD -s $RESEAU_NODE -p tcp -m multiport –dports $OKPORTS_GLOBAL_UDP -d $RESEAU_GLOBAL -j ACCEPT

#On définit les règles qui seront appliquées en TCP et UDP depuis le réseau du node vers le réseau de la communauté
iptables -A FORWARD -s $RESEAU_NODE -p tcp -m multiport –dports $OKPORTS_AW_TCP -d $RESEAU_GLOBAL_AW -j ACCEPT
iptables -A FORWARD -s $RESEAU_NODE -p tcp -m multiport –dports $OKPORTS_AW_UDP -d $RESEAU_GLOBAL_AW -j ACCEPT

#On définit les règles qui seront appliquées en TCP et UDP depuis le réseau de la communauté vers le réseau du  node
iptables -A FORWARD -i $IFAW -s $RESEAU_GLOBAL -p tcp -m multiport –dports $OKPORTS_GLOBAL_TCP -d $RESEAU_NODE  -j ACCEPT
iptables -A FORWARD -i $IFAW -s $RESEAU_GLOBAL -p tcp -m multiport –dports $OKPORTS_GLOBAL_UDP -d $RESEAU_NODE -j ACCEPT

#On définit les règles qui seront appliquées en TCP et UDP depuis le réseau de l'association vers le réseau du node
iptables -A FORWARD -s $RESEAU_GLOBAL_AW -p tcp -m multiport –dports $OKPORTS_AW_TCP -d $RESEAU_NODE -j ACCEPT
iptables -A FORWARD -s $RESEAU_GLOBAL_AW -p tcp -m multiport –dports $OKPORTS_AW_UDP -d $RESEAU_NODE -j ACCEPT

#Maintenant on définit ce qui est autorisé sur le serveur, on utilisera les ports définis précédemment.
iptables -A INPUT -p tcp -m multiport –dports $NODE_PORTS_TCP -j ACCEPT
iptables -A INPUT -p udp -m multiport –dports $NODE_PORTS_UDP -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport –dports $NODE_PORTS_TCP -j ACCEPT
iptables -A OUTPUT -p udp -m multiport –dports $NODE_PORTS_UDP -j ACCEPT

#On interdit tout ce qui vient d'une IP  distribuée par le réseau en forward
iptables -A FORWARD -s 10.0.0.0/25 -i $IFNAS -j DROP
```

# Chapitre 11: Installation et mise en place d'un Serveur Radius

## A propos de ce chapitre

Dans ce chapitre, vous apprendrez à mettre en place Radius sur le réseau , qui permettra de sécuriser et identifier l'accès des utilisateurs au réseau. Cette partie est de loin la plus complexe à mettre en oeuvre mais permettra de sécuriser convenablement l'accès des utilisateurs au réseau, aussi bien au niveau contrôle d'accès au réseau qu'au niveau sécurisation des données circulant sur les ondes Radio par l'intermédiaire de matériel wireless.

## Partie 1: Fonctionnement de Radius

Radius a un fonctionnement assez simple, et relativement fiable, un système Radius est divisé en 4 parties:

- Un serveur RADIUS
- Une base de données utilisateurs (mysql dans notre cas)
- Un NAS RADIUS aussi appellé client RADIUS
- Un client final.

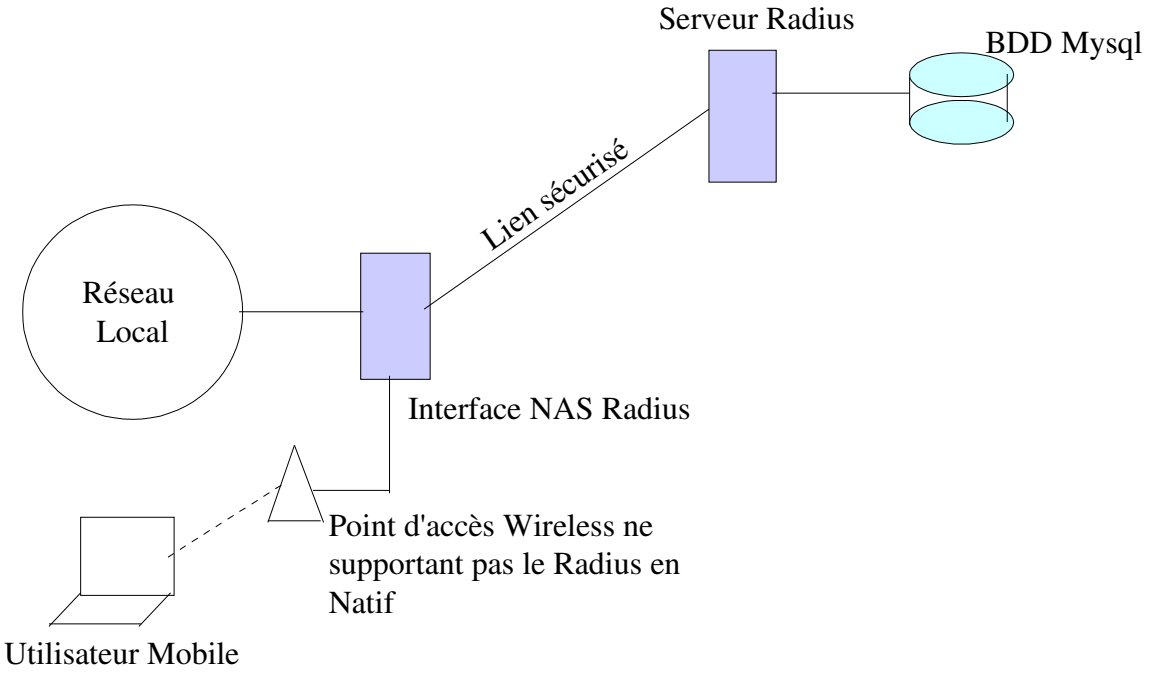
Le serveur RADIUS s'occupe de gérer la connexion des utilisateurs, il reçoit la demande provenant du NAS Radius, consulte la base de données utilisateur, et renvoie les informations de connexion au NAS Radius qui débloque l'accès de l'utilisateur si l'authentification à réussie.

La base de données utilisateurs contient la liste des utilisateurs Radius ainsi que les informations de connexion à leur renvoyer si leur tentative de connexion s'achève avec succès.
Les bases de données Mysql fonctionne par système d'utilisateurs et de groupes, il est possible d'attribuer des paramètres à un groupe d'utilisateurs ou à un seul utilisateur. Les tables Mysql seront décrites par la suite.

Le NAS (Network Accès Service) Radius est le point d'accès au réseau, il fonctionne de manière bloquante, si une authentification d'un utilisateur se réalise avec succès son adresse IP est débloquée par le NAS et il peut accéder au réseau.

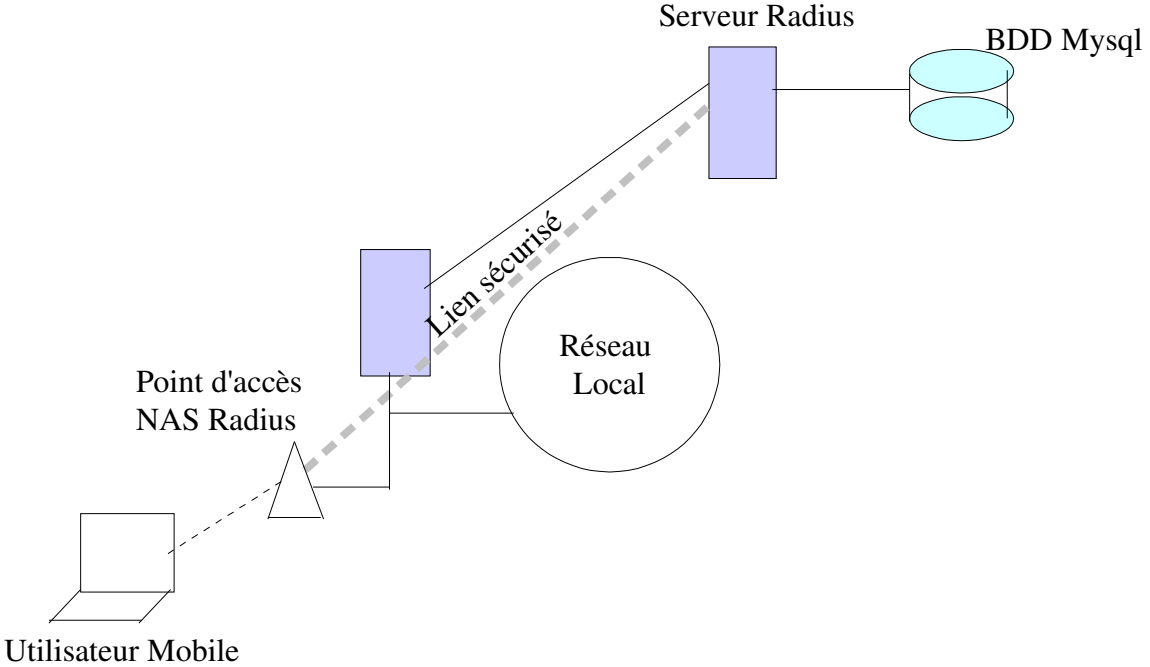
Le client final se contente d'envoyer ses informations de connexion, et de se faire débloquer son accès au réseau.

Le NAS Radius doit donc avoir sa propre interface réseau (carte ou point d'accès) pour pouvoir fonctionner, en effet l'interface du NAS Radius doit être physiquement séparée du réseau auquel on cherche à accéder.




*Chapitre 11 – Partie 1 – Schéma 1: Point d'accès sans support Radius natif*

Voici le schéma d'une installation Radius utilisant un point d'accès ne supportant pas Radius en natif, on peut tout à fait imaginer de connecter n'importe quel type de matériel réseau sur l'interface NAS Radius. Le problème majeur est le fait que Radius ne peut pas empêcher les utilisateurs de se connecter au Point d'accès, mais ils ne pourront accéder à aucune ressource du réseau.



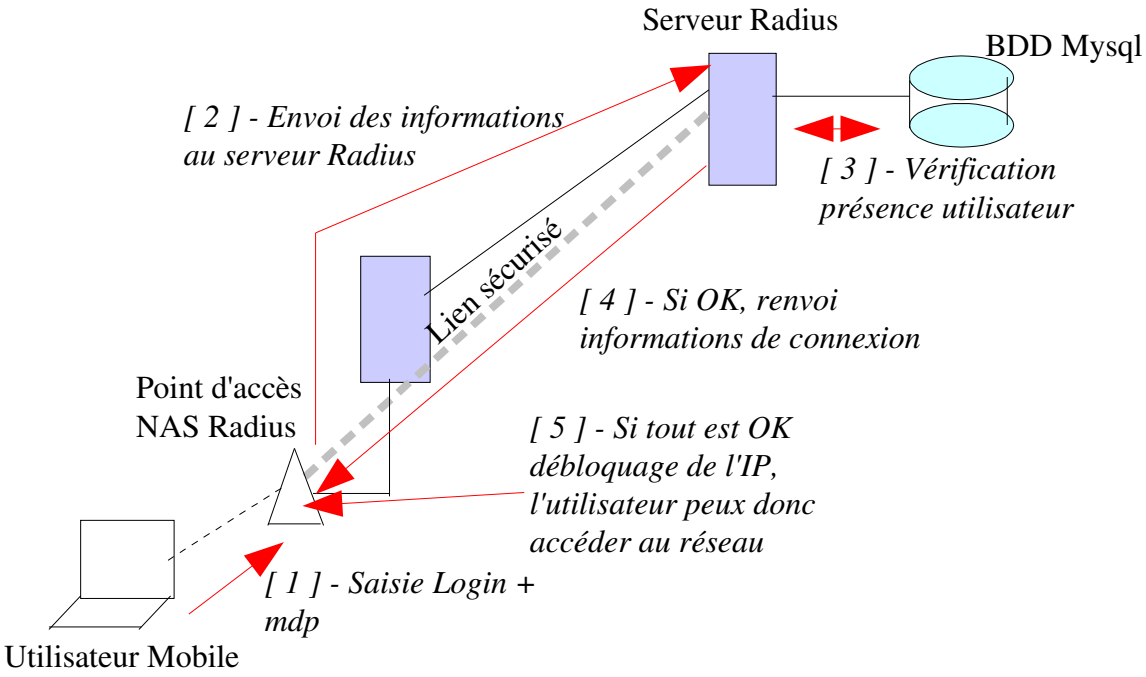

*Chapitre 11 – Partie 1 – Schéma 2: Point d'accès avec support Radius*

Un point d'accès supportant Radius peut être connecté à n'importe quel endroit du réseau, à partir du moment ou il peut dialoguer avec le serveur Radius. Le seul inconvénient qu'il peut y avoir est la liste de serveurs Radius que peux contacter le NAS qui risque d'être limité à un nombre d'adresses réduit.
Il est important de noter que les données transitant entre le NAS et le serveur Radius

sont cryptées, grâce à un échange de clef partagée (appelée Shared Key ou Secret) la communication entre le serveur et le Nas peut s'effectuer, de plus le réseau VPN étant crypté nous n'avons pas trop à nous inquiéter sur la sécurité des échanges de données entre le NAS et le Radius, le seul risque pourrait être une attaque d'une personne déjà identifiée, mais il faudrait pour cela qu'elle arrive a passer la sécurité du cryptage de données entre le NAS et le Radius, de plus la personne étant logguée, il sera très aisé de la retrouver.

Dans notre cas il y aura 2 serveurs Radius par ville, avec sur le NAS Radius une liste des serveurs à contacter en cas d'échec sur l'identification d'un utilisateur.

Voici le schéma d'une procédure de connexion d'un utilisateur.




*Chapitre 11 – Partie 1 – Schéma 3: Point d'accès avec support Radius*

Nous voyons ici de quelle manière se déroule l'authentification d'un utilisateur au réseau. Les points d'accès supportant Radius en natif étant relativement rares, nous n'aborderons leur installation dans une prochaine version de ce dossier.

## Partie 2: Installation de ppp et pptp

Nous allons maintenant voir comment se déroulera la connexion d'un utilisateur mobile sur le réseau. Quand il se connectera, il se fera attribuer une plage d'adresses Ips par le serveur DHCP, avec cette plage il ne pourra pas communiquer avec le reste du réseau, il sera bloqué au niveau du point d'accès. Par contre lorsqu'il essayera de se connecter, si sa connexion est validée par le serveur radius, il recevra par pptp des ips lui permettant de communiquer avec le reste du réseau et d'accéder à internet.
Pptp est un protocole de tunneling (vpn) standard et supporté en natif par Windows (2K et XP)

```
1:# cd /usr/src/linux/ppp
2:# sh configure ; make; make install
3:# apt-get install pptpd
```

Ligne 1: Déplacement dans le repertoire du logiciel ppp
Ligne 2: Installation de ppp
Ligne 3: Installation de pptpd

Configuration de pptpd, dans ce fichier, nous allons définir la plage d'adresses ips qui sera attribuée aux clients.

*Fichier /etc/pptpd.conf*

```
Option /etc/ppp/pptpd-options
localip 10.49.1.1
remoteip 10.49.1.2-14
```

Dans ce fichier sont définis les paramètres concernant la connexion d'un utilisateur au réseau.

*Fichier /etc/ppp/pptpd-options*

```
login
debug
kdebug 4
logfile /var/log/ppp-pptp
name pptpd
mru 1396
proxyarp
asyncmap 0
-chap
-mschap
+mschap-v2
domain angers.fw
ms-dns 10.49.1.1
ms-wins 10.49.1.1
netmask 255.255.255.240
lcp-echo-failure 30
lcp-echo-interval 5
ipcp-accept-local
ipcp-accept-remote
plugin radius.so
```

Une fois le serveur radius installé, le serveur sera en mesure d'accepter des connexions provenant de clients extérieurs.
Pour lancer le serveur, faites

```
1:# pptpd -l 10.0.0.1
```

Ligne 1: Lancement de pptpd en écoute sur l'adresse 10.0.0.1

## Partie 3: Installation de FreeRadius avec support Mysql

```
1:# apt-get install libmysqlclient10-dev
2:# wget ftp://ftp.freeradius.org/pub/radius/freeradius-0.9.3.tar.gz
3:# tar xvfz freeradius-0.9.3.tar.gz
4:# cd freeradius-0.9.3
5:# ./configure --with-mysql-include-dir=/usr/include/mysql  --with-
mysql-lib-dir=/var/lib/mysql --with-mysql-dir=/var/lib/mysql --
sysconfdir=/etc
6:# make; make install
```

Ligne 1: Installation des fichiers de développement de mysql (nécessaires à la compilation de freeradius)
Ligne 2: Téléchargement de l'archive du logiciel
Ligne 3: Décompression de l'archive.
Ligne 4: Déplacement dans le répertoire du logiciel.
Ligne 5: Préparation du fichier permettant la compilation, ici nous indiquons l'emplacement de mysql et des fichiers de configuration.
Ligne 6: Installation de freeradius

## Partie 4: Configuration de FreeRadius

Nous allons maintenant configurer FreeRadius pour fonctionner avec Mysql.
Nous allons tout d'abord réaliser des tests en local en configurant le serveur pour accepter les connexions locales (le serveur fera office de NAS)

*Fichier /etc/raddb/clients.conf*

```
client 127.0.0.1 {
        secret = xxxxxx
        shortname = localhost
        nastype = other     # localhost isn't usually a NAS...
}
client 10.49.1.80/28 {
        secret = xxxxxx
        Shortname = reseau-gwe
}
```

Le secret doit être identique sur le fichier clients.conf à la clef du fichier clients
Ce fichier contient les adresses des différents réseaux ou sont situés les NAS (les réseaux des différents noeuds) et les clefs de secret (nécessaires au cryptage).

*Fichier /etc/raddb/clients*

```
# Client Name         Key
#----------------      ----------
localhost           xxxxxx
reseau-gwe          xxxxxx
```

Noms des clients locaux, ce fichier sera utile lors de l'installation de NAS.

*Fichier* /usr/local/etc/raddb/realms

```
# Realm              Remote server [:port]      Options
#----------------    ---------------------      -------
DEFAULT              LOCAL
```

Fichier /usr/local/etc/raddb/radiusd.conf

```
log_auth = yes (ligne 249)
proxy_requests = yes (ligne 370)
sql (ligne 1445)
sql (ligne 1560)
```

Fichier /usr/local/etc/raddb/sql.conf

```
# Connect info
server = "localhost"                #Adresse du serveur mysql
login = "root"                      #Utilisateur
password = "rootpass"               #Mot de passe
# Database table configuration
radius_db = "radius"                #Nom de la base de données
```

Nous allons maintenant créer et peupler la base de données Mysql.

Tout d'abord connectez vous sur l'interface web phpMyAdmin et créez une base "radius" (ou un autre nom).

Puis dans cette base nous allons importer le schéma nécessaire à radius pour fonctionner.
Ouvrez une nouvelle console, placez vous dans un répertoire de votre choix.

```
1:# sftp root@Adresse_serveur
root@Adresse_serveur's password:
2:# cd
/rep_freeradius_du_serveur/src/modules/rlm_sql/drivers/rlm_sql_mysql/
3:# get db_mysql.sql
4:# bye
```

Revenez à l'interface phpMyAdmin puis dans "SQL" cliquez sur "Browse" et allez chercher sur votre disque le fichier db_mysql.sql

*Your SQL-query has been executed successfully*

**Fonctionnement des tables Mysql**

La base de données mysql de freeradius est composée de 6 tables permettant de bien définir les droits de chaque utilisateur, de la même manière que pour des comptes d'utilisateurs classiques (novell, ldap, nis, ...), mais avec des droits spécifiques à radius.

### Table usergroup

Cette table permet de définir les utilisateurs et leur appartenance à un groupe. Il sera en effet possible de définir des droits qui seront appliqués automatiquement à tous les utilisateurs d'un groupe, pour simplifier les tâches d'administrations.

| Id | Username | groupname |
|---|---|---|
| 1 | fanfsql | test |

### Table radcheck

Cette table contient les mot de passe des différents utilisateurs

| Id | UserName | Attribute | Value | Op |
|---|---|---|---|---|
| 1 | fanfsql | Password | **xxxxxx** | = = |

### Table radgroupcheck

Cette table contient les différents groupes

| Id | GroupName | Attribute | Value | Op |
|---|---|---|---|---|
| 1 | user | Auth-Type | Local | : = |
| 2 | node_master | Auth-Type | Local | : = |
| 3 | administrateur | Auth-Type | Local | : = |

### Table radreply

Cette table contient les paramètres qui seront renvoyés à chaque utilisateur.

| Id | UserName | Attribute | Value | Op |
|---|---|---|---|---|
| 1 | fanfsql | xxx | xxx | : = |

### Table radgroupreply

Cette table contient les paramètres qui seront renvoyés à chaque groupe d'utilisateurs.

| Id | GroupName | Attribute | Value | Op |
|---|---|---|---|---|
| 1 | test | xxx | xxx | : = |

## Partie 5: Tests en local

Nous avons maintenant un serveur configuré, nous allons faire quelques tests d'authentification en local.
Nous allons commencer par insérer un utilisateur dans la base de données Mysql.

Une fois que le table est peuplée, nous allons faire un test de connexion en

local.
Commencez par lancer le serveur en mode debug

```
1:# radiusd -X
Going to the next request
--- Walking the entire request list ---
Waking up in 6 seconds...
--- Walking the entire request list ---
Cleaning up request 2 ID 129 with timestamp 3feeb0cb
Nothing to do.  Sleeping until we see a request.
```

Ligne 1: Lancement du serveur radius en mode debug

Vous pouvez constater que le serveur se met en attente d'une nouvelle connexion, ouvrez une nouvelle console (Alt + F2) et lancez une connexion au serveur.

```
1:awserver:~# radtest fanfsql Mdputil 127.0.0.1 1812 Mdpkey
Sending Access-Request of id 229 to 127.0.0.1:1812
        User-Name = "fanfsql"
        User-Password = "Mdputil"
        NAS-IP-Address = awserver
        NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=229, length=44
        Framed-Protocol = PPP
        Service-Type = Framed-User
        Framed-Compression = Van-Jacobson-TCP-IP
```

Ligne 1: Lancement d'une demande de connexion d'un utilisateur

Vous pouvez constater que le serveur reçoit bien les demandes de connexion.

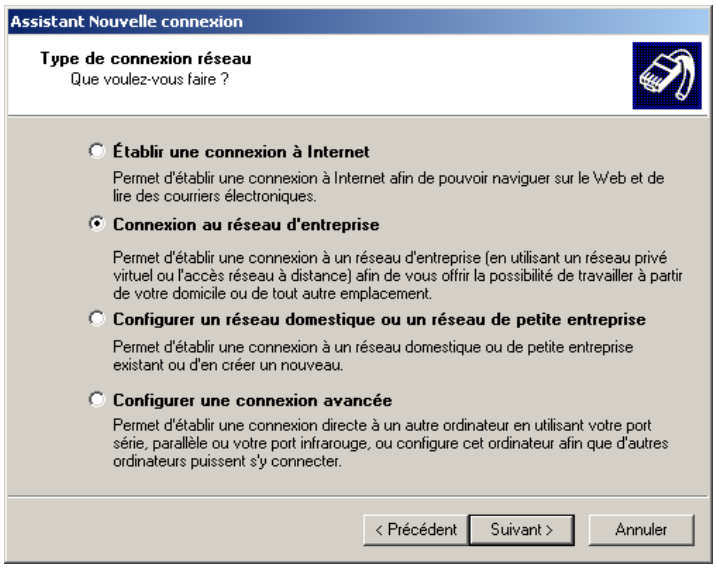
```
rad_recv: Access-Request packet from host 127.0.0.1:35796, id=124, length=59
        User-Name = "fanfsql"
        User-Password = "Mdputil"
        NAS-IP-Address = 255.255.255.255
        NAS-Port = 1812
... #les lignes Mysql ont étés supprimées pour un gain de temps.
Sending Access-Accept of id 124 to 127.0.0.1:35796
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
Finished request 0
Going to the next request
--- Walking the entire request list ---
Waking up in 6 seconds...
--- Walking the entire request list ---
Cleaning up request 0 ID 124 with timestamp 3ff30edd
Nothing to do.  Sleeping until we see a request.
```

Ligne 1: Lancement du serveur radius en mode debug

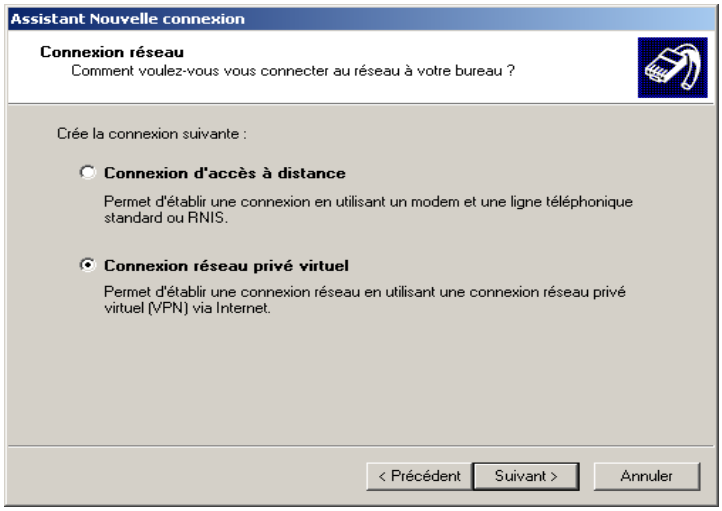
## Partie 6: Connexion d'un utilisateur au réseau.

Nous allons maintenant réaliser des tests en conditions réelles, nous allons connecter un utilisateur mobile au réseau.

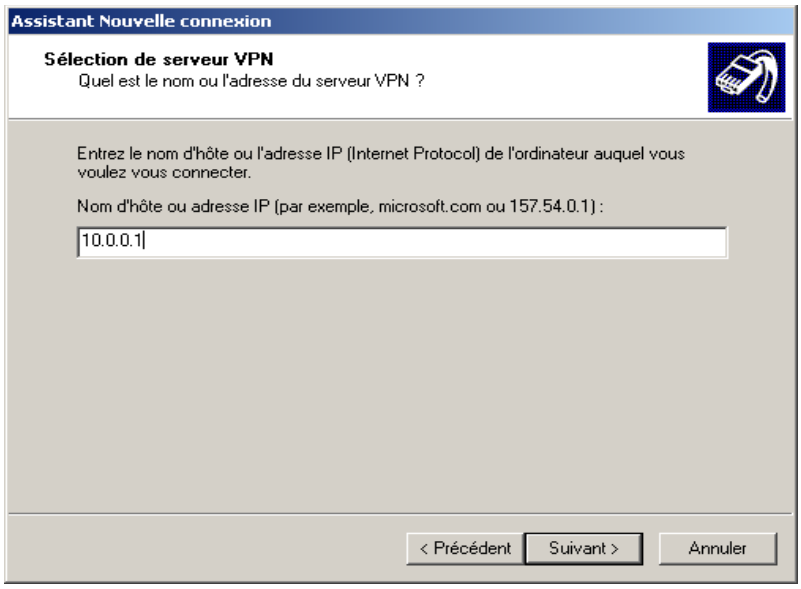
Pour configurer un poste sous client sous windows, sélectionnez "ajouter une nouvelle connexion", puis "Connexion au réseau d'entreprise"



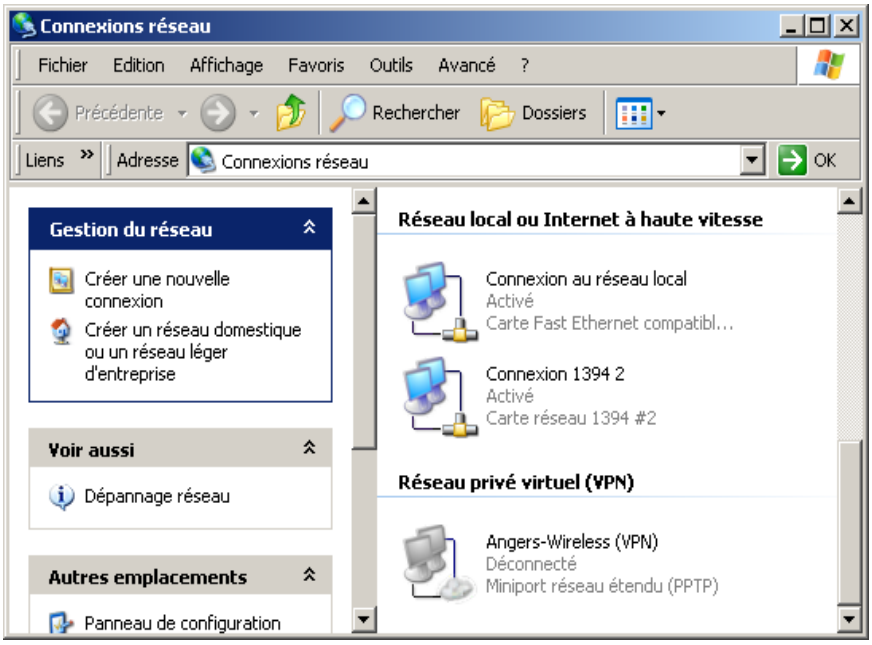

sélectionnez ensuite "Connexion au réseau privé virtuel"



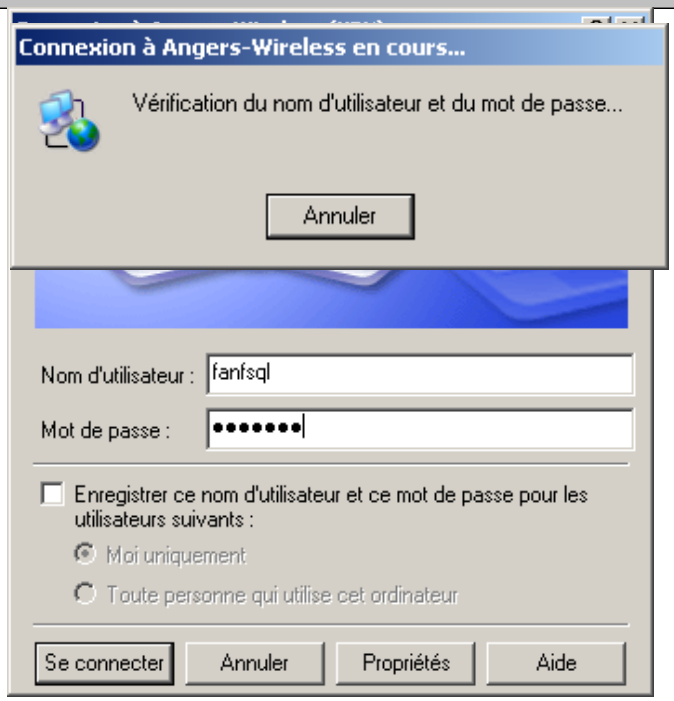
Indiquez ensuite l'adresse IP du serveur pptp.




Vous pouvez constater la présence d'une nouvelle connexion VPN (pptp) dans la fenêtre affichant les connexions réseau.



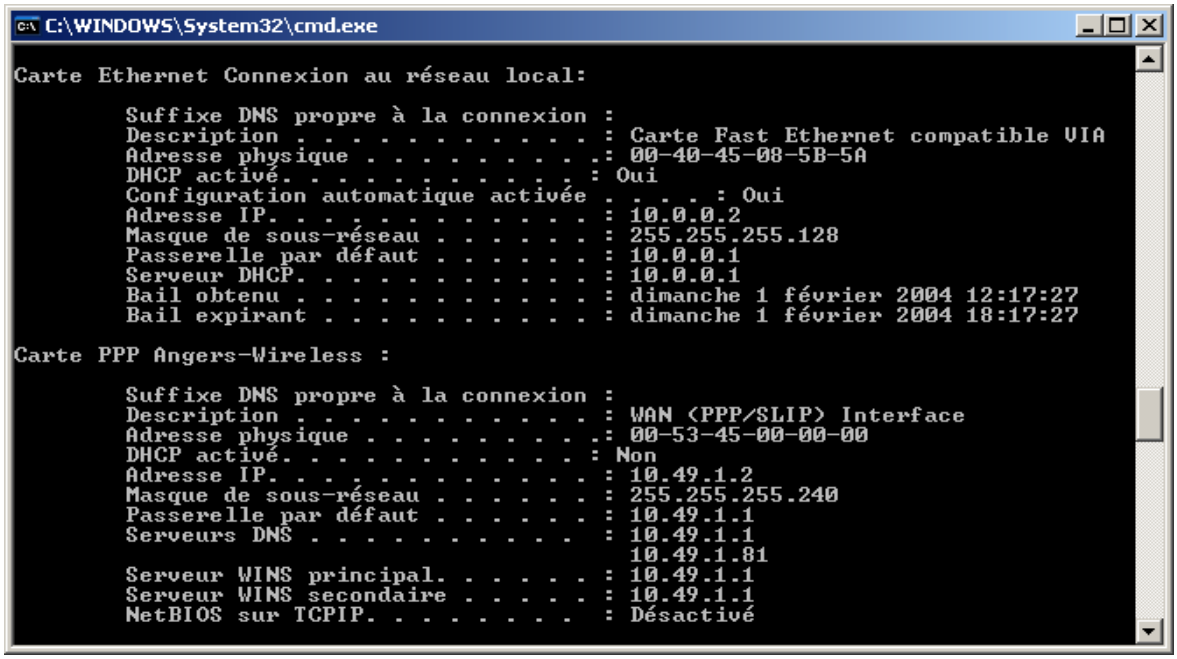

En double-cliquant sur l'icone du réseau, vous obtenez une fenêtre qui vous demande votre login et mot de passe.





Une fois connecté, vous pouvez constater qu'une nouvelle connexion apparait au niveau de la barre de taches.

Si vous faites un ipconfig dans une console, vous pouvez constater qu'il y a 2 connexions.



Au niveau du serveur RADIUS, nous pouvons constater que ce dernier à bien reçût une demande de connexion.

```
Ready to process requests.
rad_recv: Access-Request packet from host 127.0.0.1:13976, id=113, length=74
        User-Name = "fanfsql"
        User-Password = "\025\034U\2707\366\002\324\236\375u?wa\002Y"
        NAS-IP-Address = 127.0.0.1
        NAS-Identifier = "ppp"
        NAS-Port = 12951
        NAS-Port-Type = Virtual
        Service-Type = Authenticate-Only
modcall: entering group authorize for request 0
```

Et validé cette demande de connexion

```
Sending Access-Accept of id 113 to 127.0.0.1:13976
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
Finished request 0
Going to the next request
--- Walking the entire request list ---
Waking up in 6 seconds...
--- Walking the entire request list ---
Cleaning up request 0 ID 124 with timestamp 3ff30edd
Nothing to do.  Sleeping until we see a request.
```

# Chapitre 12: Script de démarrage

Nous allons dans cette partie créer un petit script permettant de lancer automatiquement certains services au démarrage du système. Nous avons vu tout au long de ce dossier que de nombreux logiciels doivent être démarrés au lancement du système, nous allons donc créer un script adapté.

*Fichier start-server*

```
/root/script-iptables &
vtund -s
vtund AW-Lolo loloserv.ath.cx #Lancement des différents tunnels
vtund AW-Gwe gweserv.ath.cx
zebra &
ospfd &
bgpd &
radiusd -i 10.0.0.1 &
pptpd -l 10.0.0.1 &
```

Maintenant nous allons faire en sorte que ce script se lance à chaque démarrage de la machine au runlevel 3.

```
1:# update-rc.d start-server defaults 03
```

Ligne 1: Création d'un script se lancant a chaque démarrage de la machine au runlevel 3

Si vous désirez supprimer ce script faites "update-rc.d start-server remove"

# Chapitre 13: Tableau d'attribution réseau

Pour vous aider dans votre création du réseau, vous pouvez remplir ce formulaire:

| Pseudo | Adresse réseau | Masque réseau | Broadcast | Adresse serveur node | Plage IP | |
|---|---|---|---|---|---|---|
| AW | 10.xx.1.0 | 255.255.255.240 | 10.xx.1.15 | 10.xx.1.1 | 2 à 14 | |
| teuxe | 10.xx.1.16 | 255.255.255.240 | 10.xx.1.31 | 10.xx.1.17 | 18 à 30 | |
| lessyv | 10.xx.1.32 | 255.255.255.240 | 10.xx.1.47 | 10.xx.1.33 | 34 à 46 | |
| onyme | 10.xx.1.48 | 255.255.255.240 | 10.xx.1.63 | 10.xx.1.49 | 50 à 62 | |
| xxx | 10.xx.1.64 | 255.255.255.240 | 10.xx.1.79 | 10.xx.1.65 | 66 à 78 | |
| gwe | 10.xx.1.80 | 255.255.255.240 | 10.xx.1.95 | 10.xx.1.81 | 82 à 94 | |
| quinq | 10.xx.1.96 | 255.255.255.240 | 10.xx.1.111 | 10.xx.1.97 | 98 à 110 | |
| LoLo | 10.xx.1.112 | 255.255.255.240 | 10.xx.1.127 | 10.xx.1.113 | 114 à 126 | |
| | 10.xx.1.128 | 255.255.255.240 | 10.xx.1.143 | 10.xx.1.129 | 130 à 142 | |
| | 10.xx.1.144 | 255.255.255.240 | 10.xx.1.159 | 10.xx.1.145 | 146 à 158 | |
| | 10.xx.1.160 | 255.255.255.240 | 10.xx.1.175 | 10.xx.1.161 | 162 à 174 | |
| | 10.xx.1.176 | 255.255.255.240 | 10.xx.1.191 | 10.xx.1.177 | 178 à 190 | |
| | 10.xx.1.192 | 255.255.255.240 | 10.xx.1.207 | 10.xx.1.193 | 194 à 206 | |
| | 10.xx.1.208 | 255.255.255.240 | 10.xx.1.223 | 10.xx.1.209 | 210 à 222 | |
| | 10.xx.1.224 | 255.255.255.240 | 10.xx.1.239 | 10.xx.1.225 | 226 à 238 | |
| | 10.xx.1.240 | 255.255.255.240 | 10.xx.1.255 | 10.xx.1.241 | 242 à 254 | |
| | 10.xx.2.0 | 255.255.255.240 | 10.xx.2.15 | 10.xx.2.1 | 2 à 14 | |
| | 10.xx.2.16 | 255.255.255.240 | 10.xx.2.31 | 10.xx.2.17 | 18 à 30 | |
| ... | ... | ... | ... | ... | ... | ... |

# Chapitre 14: Lexique

| Mot | Définition |
|---|---|
| Allias | Raccourcis, un poste réseau peux avoir 2IP réseau et une seule carte réseau, une des 2 ips est donc un allias de la première. |
| AP | Voir Point d'accès Wireless. |
| Apache | Serveur internet. |
| BGP4 | Protocole de routage dynamique. |
| Broadcast | Adresse de diffusion d'un réseau, correspond aussi à la dernière adresse IP d'un réseau (192.168.1.255 par exemple). |
| Classe A | Adresses IP allant de 1.0.0.0 à 126.0.0.0. |
| Classe B | Adresses IP allant de 128.0.0.0 à 191.255.0.0. |
| Classe C | Adresses IP allant de 192.0.0.0 à 223.255.255.0. |
| Cvs | Logiciel permettant de récupérer des répertoires sur des serveurs distants. |
| Debian | Distribution Linux. |
| Desktop | Ordinateur de bureau en anglais. |
| DHCP | Dynamic Host Configuration Protocol, système d'attribution automatique d'adresses IP et autres paramètres réseaux. |
| Distribution | Les différents systèmes linux sont fournis sous formes de distributions, les principales différences entre ces distributions sont l'interface d'installation et les logiciels fournis. |
| DMZ | Zone démilitarisée, permet lorsqu'on possède un routeur internet de considérer un ordinateur du réseau local comme ordinateur connecté sur internet, toutes les requêtes provenant d'internet seront redirigées vers lui. |
| DNS | Domain Name Service, service permettant de faire le lien entre un nom d'ordinateur et une adresse IP, voir dossier. |
| Firewall | Pare feu, permet de protéger un ordinateur ou un réseau d'attaques extérieures ou intérieures. |
| Forwader | Voir mapping. |
| Gateway | Passerelle, serveur par lequel passeront les données pour aller à une certaine destination. |
| Hub | Boitier permettant l'inter-connexion d'équipements en réseau filaire. |
| IM | Instant Messenger, systeme de messagerie instantanée . |
| IP | Adresse d'un ordinateur sur le réseau, elle peux être fixe ou attribuée à chaque démarrage. |
| Jabber | Protocole de communication pour IM (Instant Messenger). |
| Laptop | Ordinateur portable en anglais. |
| Linux | Système d'exploitation performant, fiable, libre et gratuit. |
| Mapping | Lorsque les requêtes arrivent sur un port d'un poste (ou routeur) elles sont automatiquement redirigées vers un autre poste sur le même port ou sur un autre port. |
| Mppe | Protocole de cryptage de données. |
| Mysql | Base de données fréquemment utilisée sur internet. |
| NAS | Network Access Service. Serveur faisant office de client Radius et validant ou non les demandes de connexion qu'il reçoit. |
| Netmask | Masque de sous réseau. |
| Network | Adresse IP d'un réseau, première adresse d'une plage d'adresses, inutilisable par un poste (192.168.1.0 par exemple). |
| Node | Aussi appelé noeud, dans notre cas, un des points d'inter-connexion du réseau. |
| Noyau | Coeur du système. |
| Ospf | Protocole de routage dynamique. |
| PC | Personnal Computer, ordinateur. |
| Php | Langage de programmation, très utilisé sur internet. |
| Ping | Méthode permettant de tester la réponse d'un poste distant. |
| Point d'accès Wireless | Fonctionne de manière similaire à un hub mais pour les réseau sans fils. |
| Port | "porte" permettant d'accéder à l'ordinateur, chaque service réseau à un port par défaut de défini (http => 80 par exemple). |
| Ppp | Point to point protocol, protocole de liaison point à point, utilisé par les modems ou les connections VPN par exemple. |
| Pptp | Point to point tunneling protocole. Tunneling de type VPN supporté en natif par windows. |
| Radius | Serveur validant l'accès au réseau. |
| Radius | Protocole d'authentification à un réseau. (voir dossier) |
| Routage | Technique expliquée dans ce dossier permettant à un poste de savoir comment en trouver un autre sur un réseau étendu. |
| Routeur | Matériel permettant d'interconnecter des réseau (dans ce dossier principalement Internet <-> Réseau Local). |
| Shell | Fenêtre ou des commandes peuvent êtres éxécutés. |
| Ssh | Secure Shell, méthode permettant d'administrer un serveur à distance de manière sécurisée. |
| Switch | Matériel "intelligent" permettant d'inter-connecter des équipements en réseau. |
| VPN | Virtual Private Network, réseau privé virtuel, technique permettant d'interconnecter deux réseaux par l'intermédiaire d'internet. |
| Vtun | Logiciel de tunneling. |
| Wifi | Terme commercial pour Wireless. |
| Wins | Window Information Name Service. Sous windows permet la résolution IP <-> nom machine. |
| Wireless | Terme désignant les équipement de réseaux sans fils par ondes radio (normes 802.11x). |
| Woody | Distribution Linux Debian 3.0. |
| Zebra | Logiciel permettant de mettre en place du routage dynamique. |

# Chapitre 18: GNU Free Documentation licence.

**Table des matières**



Version 1.2, November 2002



## 1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with

the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warrany Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## 12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.