Chapitre III Mise en place d'une sécurité basée sur le 802.1x et un serveur d'authentification

Introduction

Pour la réalisation de ce projet, il a fallu mettre en place un réseau test, ceci a nécessité la mise en place d'un serveur d'authentification radius, et d'un mécanisme de génération de certificats. Nous avons opté pour l'installation de ces outils dans un environnement LINUX, d'une part parce qu'ils sont en Open Source, et d'autre part, ils sont moins vulnérables aux attaques.

- **Systèmes d'exploitation utilisés**

- Linux mandriva 2010.2 pour le serveur
- Windows XP pour les postes clients

Dans ce chapitre, on va détailler les étapes de l'installation des programmes nécessaires à notre expérimentation.

1. Installation et configuration d'OpenSSL

1.1. Installation

On a utilisé la version openssl-0.9.7g téléchargé sur le site www.openssl.org

On commence par la décompression du fichier pour l'installer en utilisant la commande suivante :

```
tar zxvf openssl-0.9.7g.tar.gz
cd openssl-0.9.7g
./config --prefix=/usr/local/openssl-certgen shared
make
make install
```

Openssl se compile, cela dure plus ou moins longtemps suivant la machine utilisée. Une fois la compilation terminée, un message comme ci-dessous s'affichera.

Figure III.1 : Compilation d'Openssl

1.2. Configuration

Il faut maintenant éditer le fichier de configuration d'openssl. Ce fichier contient différentes informations comme : le nom de l'entreprise, le pays, l'adresse e-mail, le nom du propriétaire du certificat…

L'Edition via l'éditeur de texte (nous utiliserons **gedit**) du fichier de configuration openssl.cnf

```
gedit /usr/local/openssl-certgen/ssl/openssl.cnf
```

Vers le milieu du fichier se trouve les paramètres à modifier : toutes les lignes qui sont de la forme XXX_default (Comme encadré ci-dessous) :

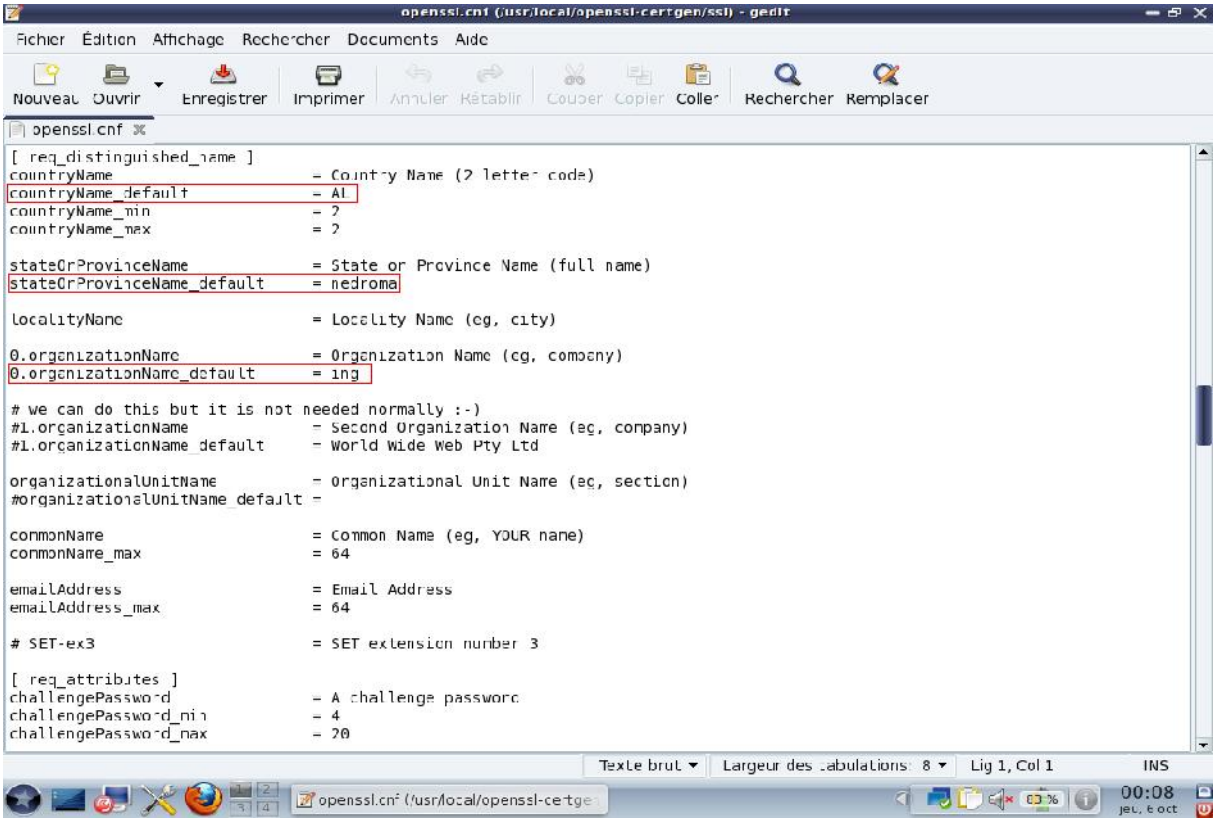

Figure III.2 : Les informations à remplir sur SSL

L'installation d'openssl est terminée.

2. Générations des certificats

Sur le site ***http://www.nantes-wireless.org/actu/article.php3?id_article=8***, nous pouvons trouver les scripts suivants : **xpextensions**, **CA.root**, **CA.svr**, **CA.clt**

Ceux-ci nécessaires à la génération des certificats.

Possédant déjà les scripts, il nous reste seulement à les copier dans le chemin approprié :

/usr/local/openssl-certgen/ssl

Attention à ne pas oublier de copier le fichier **xpextensions** contenant les OID pour la génération des certificats.

2.1. Génération du certificat root

Le certificat root lui même autorité de certification sera générer par le fichier CA.root, permettant aussi la signature des autres certificats (client, serveur,…).

Le lancement du certificat root se fera par la commande suivante :

```
[/usr/local/openssl-certgen/ssl] # chmod 700 CA.root
[/usr/local/openssl-certgen/ssl] # ./CA.root
```

A chaque question appuyée sur la touche entrer. Une fois cette série terminée (questions), la création des fichiers **root.pem**, **root.der**, **root.p12** et dossier demoCA se fera d'elle-même (dans le chemin: **/usr/local/openssl-certgen/ssl**). Le fichier root.pem est utilisé par freeradius, et il faudra installer le root.der sur chaque station client.

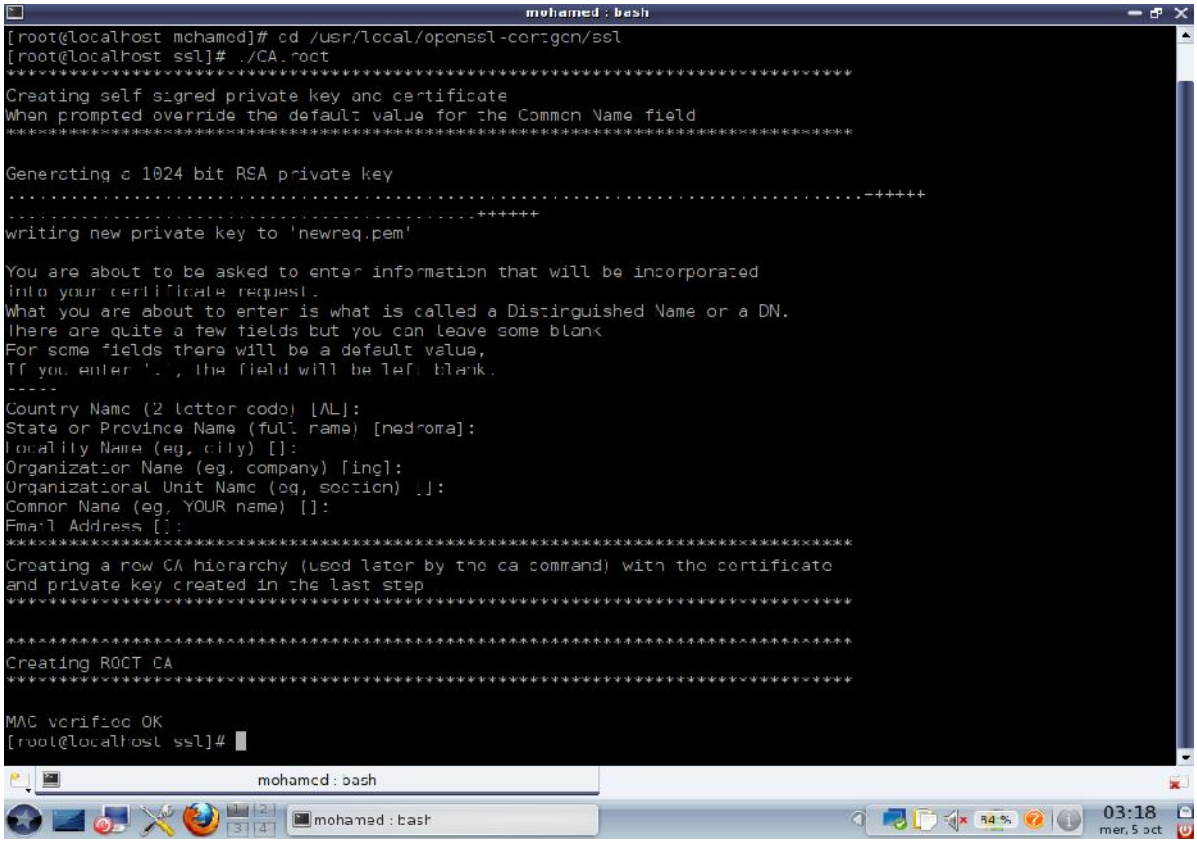

Figure III.3 : Génération du certificat root

2.2. Génération du certificat serveur

Avant d'exécuter ce script, il faut s'assurer que le fichier serial est présent dans le répertoire demoCA (crée à l'étape précédente). Dans le cas où celui-ci (serial) n'existe pas, il faudra donc le crée, puis placer une valeur hexadécimale dans ce même fichier.

A la différence du certificat root, nous devrons ajouter dans un premier temps un paramètre supplémentaire qui sera le nom du fichier que nous désirons obtenir (nom du serveur). Celui-ci devra être inscrit à la suite de l'exécution du script CA.svr comme suivant :

[/usr/local/openssl-certgen/ssl] # chmod 700 CA.svr

[/usr/local/openssl-certgen/ssl] # ./CA.svr **serveur**

Dans un second temps, il faudra répondre aux questions comme précédemment (touche entrer), ceci étant dit à la question ***Common Name (eg, YOUR name) [] :*** nous devrons répondre en utilisant le paramètre ajouté (comme ci-dessus : serveur).

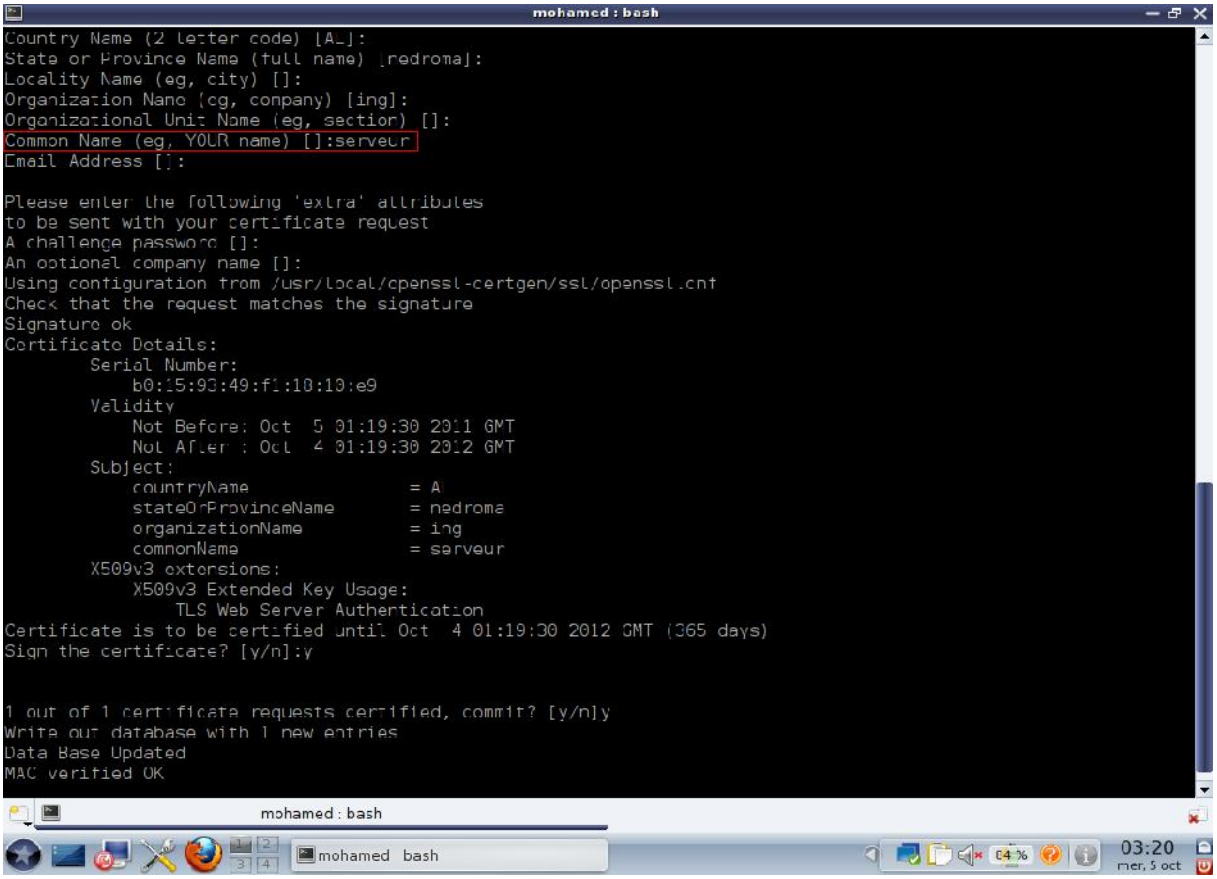

Figure III.4 : Génération du certificat serveur

On se retrouve donc avec les fichiers **serveur.pem**, **serveur.der**, **serveur.p12**, dont le dernier devra être installé sur chaque ordinateur client.

2.3. Génération du certificat client

Nous devrons réitérer la même manipulation (certificat serveur) afin d'obtenir le certificat client. Sauf qu'à la question ***Common Name (eg, YOUR name) [] :*** il faudra simplement inscrire le nom de l'utilisateur (ici se sera **wissame**) comme ci-dessous :

```
[/usr/local/openssl-certgen/ssl] # chmod 700 CA.clt
[/usr/local/openssl-certgen/ssl] # ./CA.clt wissame
```

On aura donc les 3 fichiers suivants : **wissame.pem, wissame.der, wissame.p12** dont le dernier devra être installé sur chaque ordinateur client.

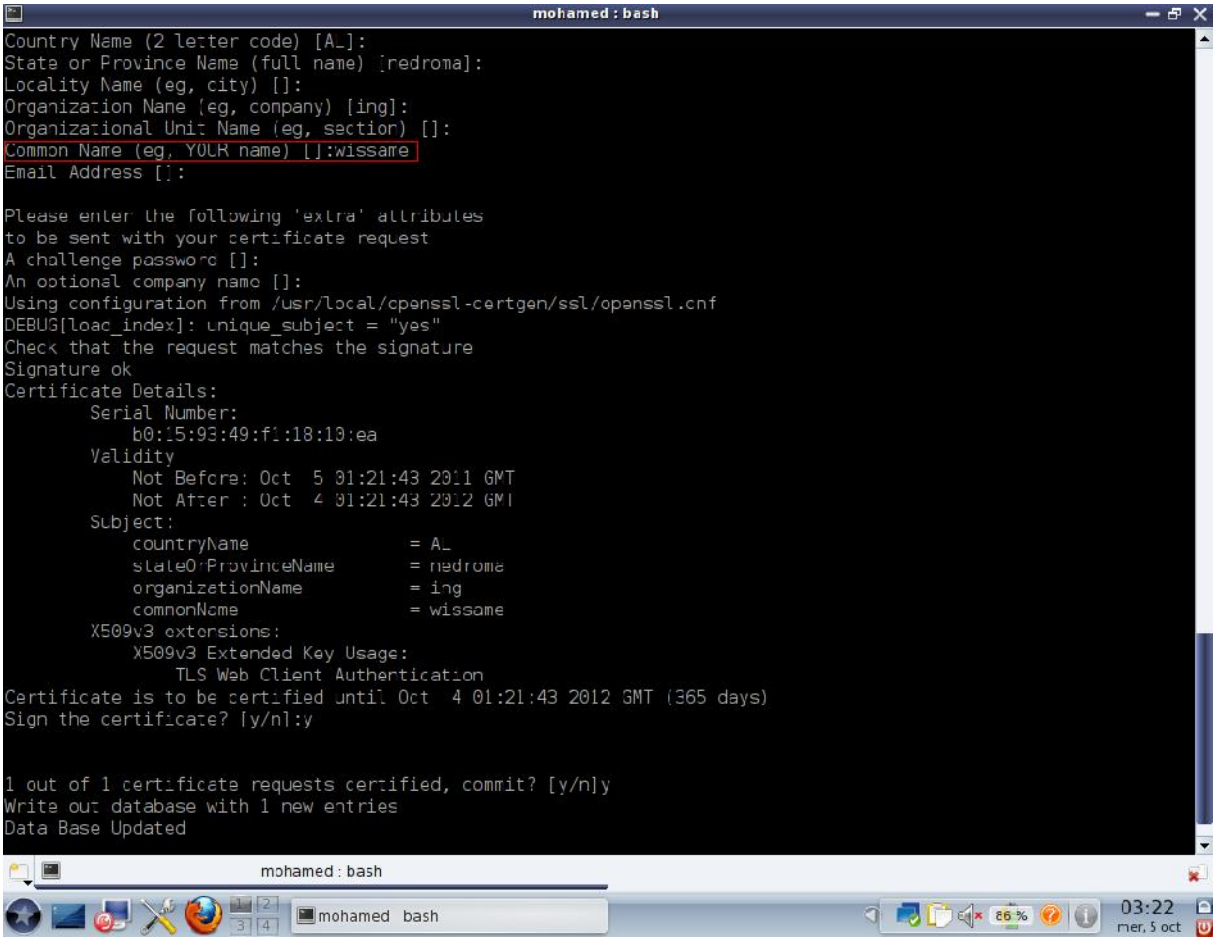

Figure III.5 : Génération du certificat client

Cependant il est impossible d'avoir 2 certificats avec le même nom d'utilisateur.

Une fois la génération des certificats est terminée, nous sommes passés à l'installation de freeradius.

3. Installation et configuration de freeradius

3.1. Installation

Version utilisé : freeradius-1.0.4 téléchargé sur le site www.freeradius.org

```
tar zxvf freeradius-1.0.4.tar.gz
cd freeradius-1.0.4
```

3.2. Configuration

Pour la configuration et la compilation de freeradius, on utilise le paramètre **--sysconfdir=/etc** qui placera tous les fichiers de configuration dans **/etc/raddb.**

```
./configure –sysconfdir=/etc
```

Important : Il faut vérifier pendant la configuration qu'il n'y a pas d'erreur au niveau d'EAP-TLS.

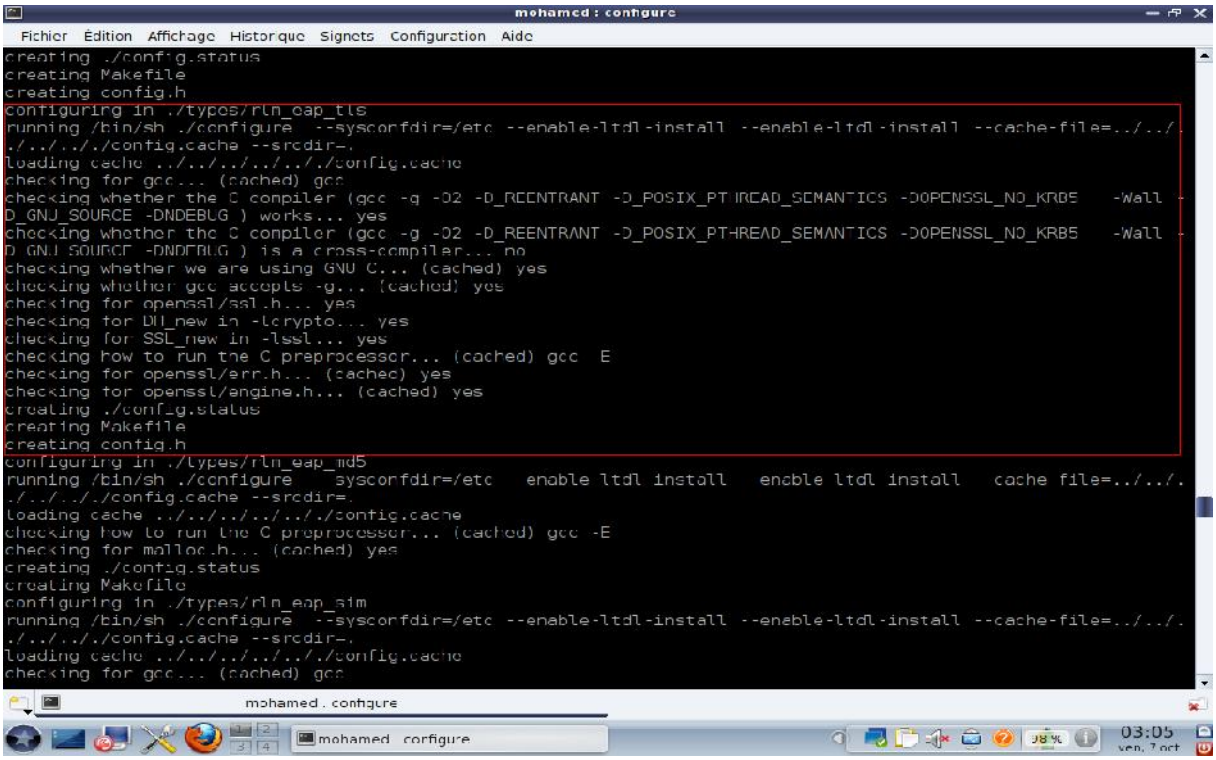

Figure III.6 : Configuration sans erreurs au niveau d'EAP-TLS

On peut passer à la compilation et l'installation de freeradius :

```
make
make install
```

Une fois l'installation terminée, un message comme ci-dessous s'affichera.

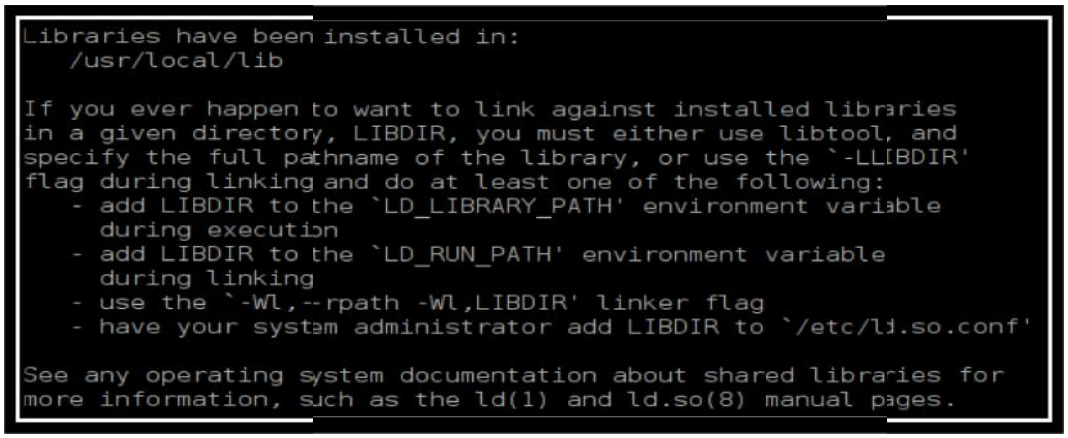

Figure III.7 : Fin d'installation de freeradius

Maintenant que freeradius est bien installé, il nous faut copier fans un première temps les certificats serveur.pem, root.pem dans le répertoire **/etc/raddb/certs** en utilisant la commande **cp**.

```
cd /etc/raddb/certs
rm –rf *
cp /usr/local/openssl-certgen/ssl/root.pem /etc/raddb/certs
cp /usr/local/openssl-certgen/ssl/serveur.pem /etc/raddb/certs
```

Dans un second temps, nous allons générer deux fichiers aléatoires : **dh** et **random**, qui vont nous permettre de mieux sécuriser notre serveur radius.

```
[/etc/raddb/cers]# openssl dhparam –check –text -5 512 –out dh
```

Enfin créez et compiler ce court programme en C pour générer un fichier comportant des caractères aléatoires.

```
[/etc/raddb/certs]# touch random.c
[/etc/raddb/certs]# gedit random.c
```

Copiez ces quelques lignes de C dans le fichier random.c :

```
#include <stdio.h>
#include <openssl/rand.h>
// you will need to compile it with openssl lib
// $ gcc –lcrypto
main void {
unsigned char buf[100] ;
if ( !RAND_bytes[buf, 100]) {
// the usual md5(time+pid)
}
Printf("Random : %s\n", buf) ;
}
```

Puis exécutez la commande suivante :

```
[/etc/raddb/certs]# gcc random.c –o random -lcrypto
```

A noter que cette commande diffère selon la version de linux.

Tester avec la commande :

```
[/etc/raddb/certs]# ./random
```

Le test donne quelque chose qui ressemble à ça :

3.3. Fichiers de configuration de freeradius

Les fichiers de configuration se trouvent dans **/etc/raddb** (comme nous l'avons précisé plus tôt via le --sysconfdir), ces fichiers sont très bien commentés et constituent la documentation de freeradius. La section suivante présente les fichiers de configuration principaux a modifié:

- **eap.conf :** pour la configuration des méthodes EAP d'authentification. Le contenu de ce fichier était au départ inclus dans la partie module du fichier « radiusd.conf » mais les développeurs ont préféré le séparer pour des raisons de lisibilité car il devenait de plus en plus volumineux du fait du nombre de méthodes d'authentification EAP différentes. En fonction des méthodes EAP que le serveur devra supporter dans son environnement de production il y aura éventuellement certains paramètres à configurer. Par exemple dans le cas d'une authentification via EAP-TLS, il faudra indiquer le répertoire contenant le certificat du serveur (qu'il enverra au supplicant) et la clé privée avec le mot de passe associé, celui contenant le certificat de l'autorité (qui permettra de vérifier le certificat fourni par le supplicant), indiquer si le serveur doit vérifier un fichier contenant les certificats révoqués ou encore s'il faut vérifier que le nom de l'utilisateur correspond au nom du propriétaire du certificat fourni.

- **clients.conf :** pour définir et paramétrer le dialogue avec les authentificateurs. Ici sont recensés les authentificateurs via un nom, une adresse IP et un secret partagé. D'autres informations optionnelles peuvent être ajoutées pour éviter les connexions simultanées d'un même utilisateur.

- **users :** est le fichier des utilisateurs. Un utilisateur est défini par son nom et sa méthode d'authentification (en fonction des méthodes, ce fichier peut contenir des mots de passe).

- **radiusd.conf :** pour la configuration globale du serveur. Ce fichier est découpé en deux grandes parties, d'abord les paramètres propres au démon (interfaces d'écoute, port, etc.), puis une partie définition des modules (définition et configuration des modules d'authentification disponibles hormis ceux du type EAP qui sont traités séparément, des modules de journalisation, de relayage des requêtes, etc.).

➢ Fichier eap.conf

```
gedit /etc/raddb/eap.conf
```

On spécifie que l'on veut utiliser EAP-TLS et non MD5
Ligne 22

```
default_eap_type = tls
```

Après on configure EAP-TLS, il faut que l'on enlève les commentaires (les # devant) a partir de la ligne 122 et on modifie les chemins des certificats :

```
tls {
        private_key_password = whatever
        private_key_file = ${raddbdir}/certs/serveur.pem
        certificate_file = ${raddbdir}/certs/serveur.pem
        CA_file = ${raddbdir}/certs/root.pem
        dh_file = ${raddbdir}/certs/dh
        random_file = ${raddbdir}/certs/random
        fragment_size = 1024
        include_length = yes
#       check_crl = yes
        check_cert_cn = %{User-Name}
}
```

private_key_password : est le mot de passe du certificat serveur (par default est whatever on peut le modifier en éditant le fichier CA.svr).
private_key_file et certificate_file : est le chemin vers le certificat serveur.
CA_file : est le chemin pour le certificat racine.
dh_file et random_file : sont les chemins vers les fichiers aléatoires qu'on a générer précédemment
check_cert_cn : permet de vérifier que le nom d'utilisateur fournit par le client est le même que celui dans le certificat (utile car certain driver propose de choisir le nom d'utilisateur et le certificat).

check_crl : est le seul paramètre qu'on laisse commenter, il permet de vérifier si le certificat n'a pas été révoqué.

- **Fichier clients.conf**

```
gedit /etc/raddb/eap.conf
```

Ce fichier permet de définir la liste des AP que l'on autorise à accéder au serveur radius. Le serveur et l'AP partagent un secret (une clé) pour crypter les données.
Par default on autorise le localhost (127.0.0.1) avec comme secret : testing123 (pour réaliser des tests en local).

```
client 127.0.0.1 {
        secret = testing123
        shortname = localhost
        nastype = other
        }
```

Pour rajoutez notre borne wifi avec comme adresse IP **192.168.1.1**

```
client 192.168.1.1 {
        secret = demoh
        shortname = D-Link
        nastype = other
          }
```

- **Fichier users**

```
gedit /etc/raddb/users
```

Éditez-le et ajoutez la ligne suivante en haut du texte, avant toute autre chose :

```
farid Auth-Type := local, User-Password == "virus"
```

Cela nous permet de vérifier les tests en local.
Dans ce fichier, on défini la liste des utilisateurs qu'on autorise. On a précédemment géré le certificat pour l'utilisateur wissame, on ajoute donc a la fin du fichier :

```
"wissame" Auth-Type := EAP
```

On spécifie que l'utilisateur « wissame » peut s'authentifier avec la méthode EAP (EAP-TLS, EAP-TTLS, EAP-PEAP,…). Pour forcer un type, il faut utiliser l'attribut EAP-Type, par exemple si on veut que l'utilisateur ne fasse que de l'EAP-TLS, il faut mettre alors :

```
"wissame"   Auth-Type := EAP, EAP-Type := EAP-TLS
```

- **Fichier radiusd.conf**

```
gedit /etc/raddb/radiusd.conf
```

Ceci étant dit la configuration de radiusd.conf ne doit pas être complètement modifiée.
Il faudra seulement s'assurer que les paramètres évoqués auparavant soient bien inscrit sur le fichier tout en respectant le modèle suivant :

```
prefix = /usr/local
exec_prefix = ${prefix}
sysconfdir = /etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/sbin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd
log_file = ${logdir}/radius.log
libdir = ${exec_prefix}/lib
pidfile = ${run_dir}/radiusd.pid
...
user = nobody
group = nogroup
...
max_request_time = 30
...
max_requests = 1024
...
bind_address = *
...
port = 0
...
hostname_lookups = yes
log_stripped_names = yes
…
log_auth = yes
…
log_auth_badpass = yes
log_auth_goodpass = yes
...
modules {
$include ${confdir}/eap.conf
}
…
authorize {         # on définit l'autorisation eap
preprocess
eap
files               # on lit le fichier users
}
authenticate {
eap                  # authentification eap
}
```

Ainsi la configuration est terminée.

- **Lancement de test du serveur**

Si tout ce passe bien, vous n'avez qu'à utiliser le daemon avec la commande :

radiusd –X –A &

On obtient à la fin :

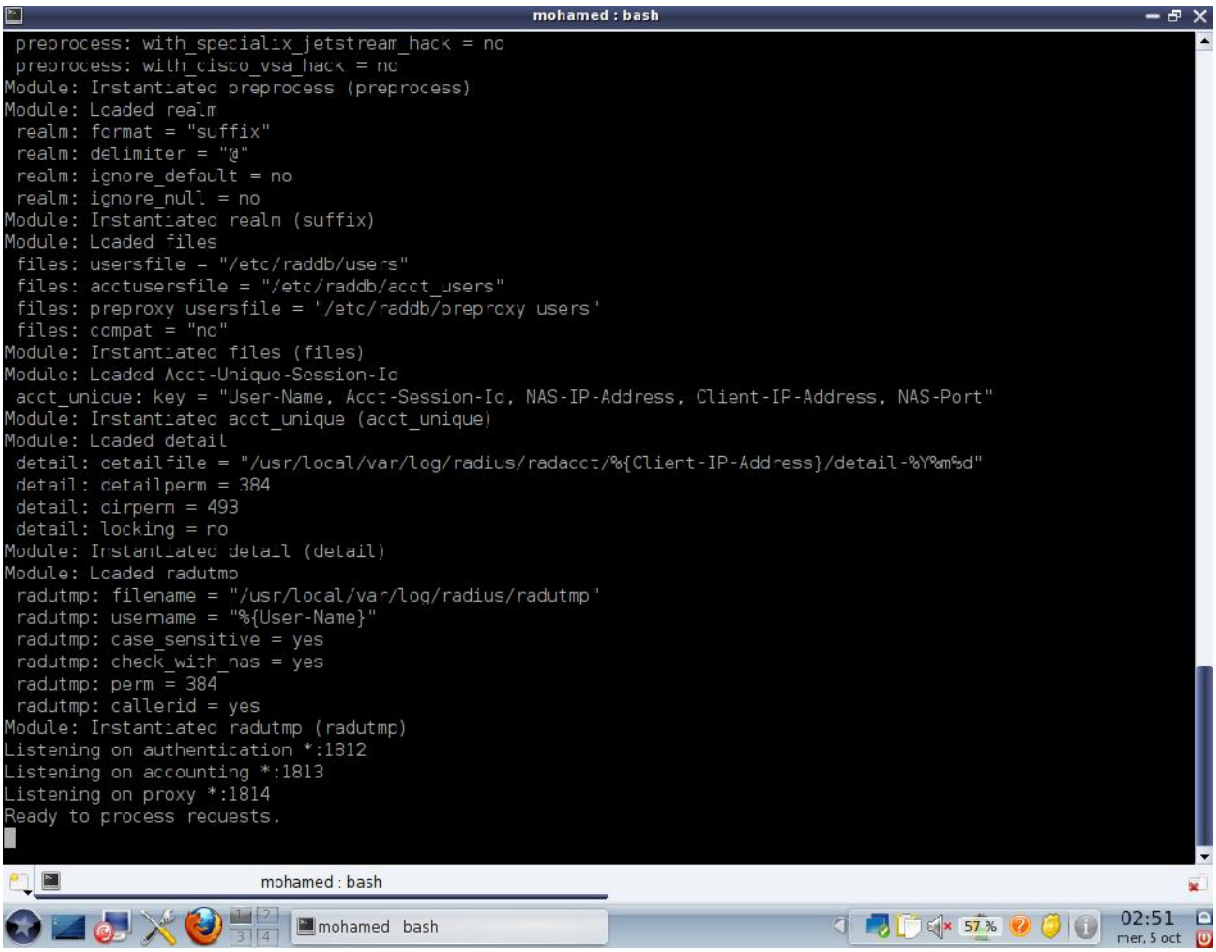

Figure III.8 : Lancement du serveur radius

Cette étape démontre que le serveur a été installé et configuré correctement.

Pour arrêter le radius, il suffit de taper :

```
killall radiusd
```

Maintenant il faut tester en local avec la commande suivante :

```
radtest farid virus localhost 0 testing123
```

On a vérifié le bon fonctionnement du serveur, par la réponse « **Access-Accept** » comme suit :

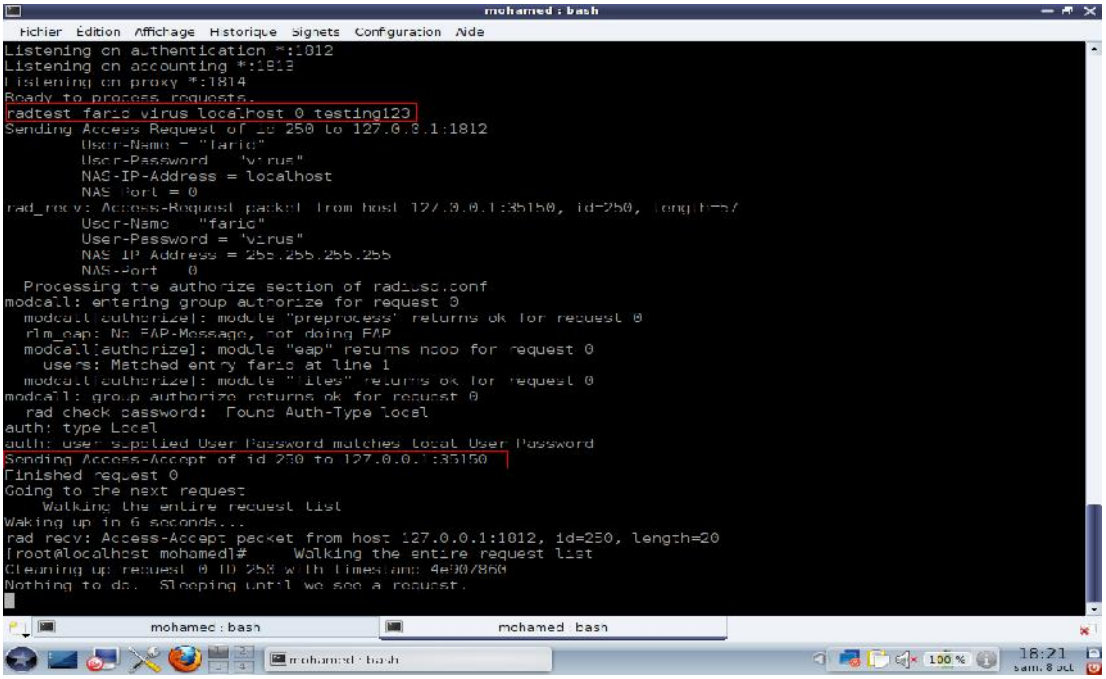

Figure III.9 : Réussite du test en local

4. Configuration du point d'accès

Le point d'accès est un **« D-Link DSL-2640U Wireless G ADSL2 + Router ».**

Voici sa configuration :

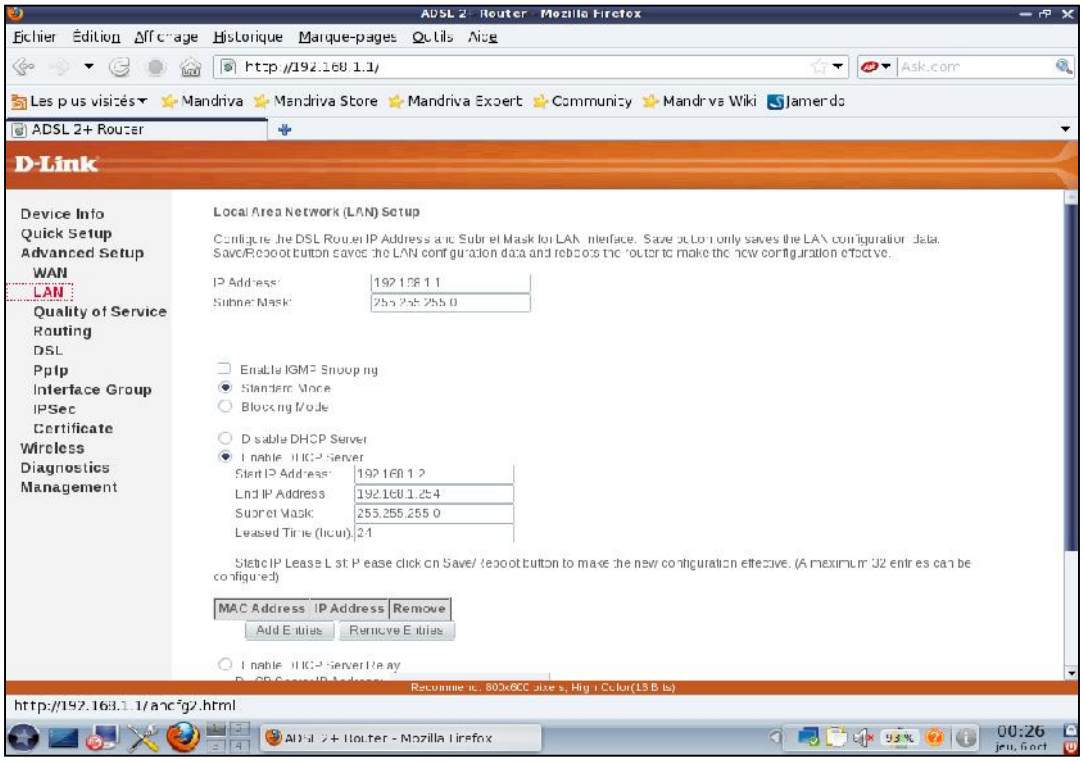

Figure III.10 : Attribution de l'adresse IP statique de l'AP

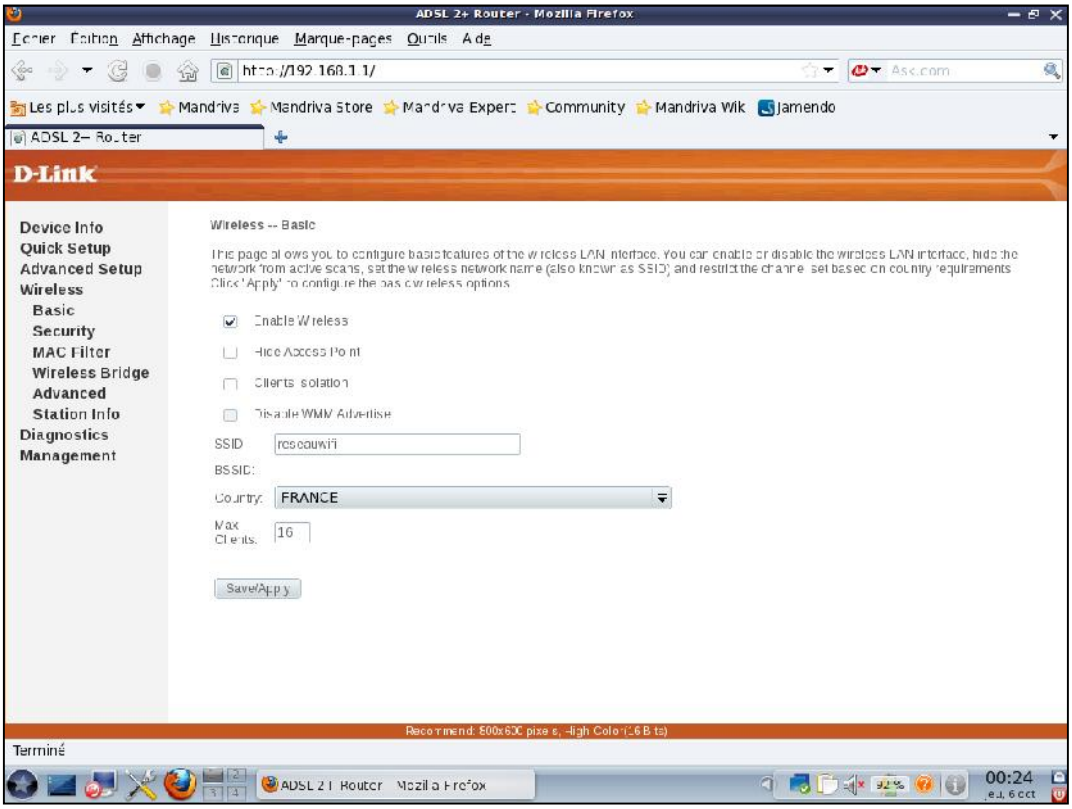

Figure III.11 : Attribution du SSID au point d'accès

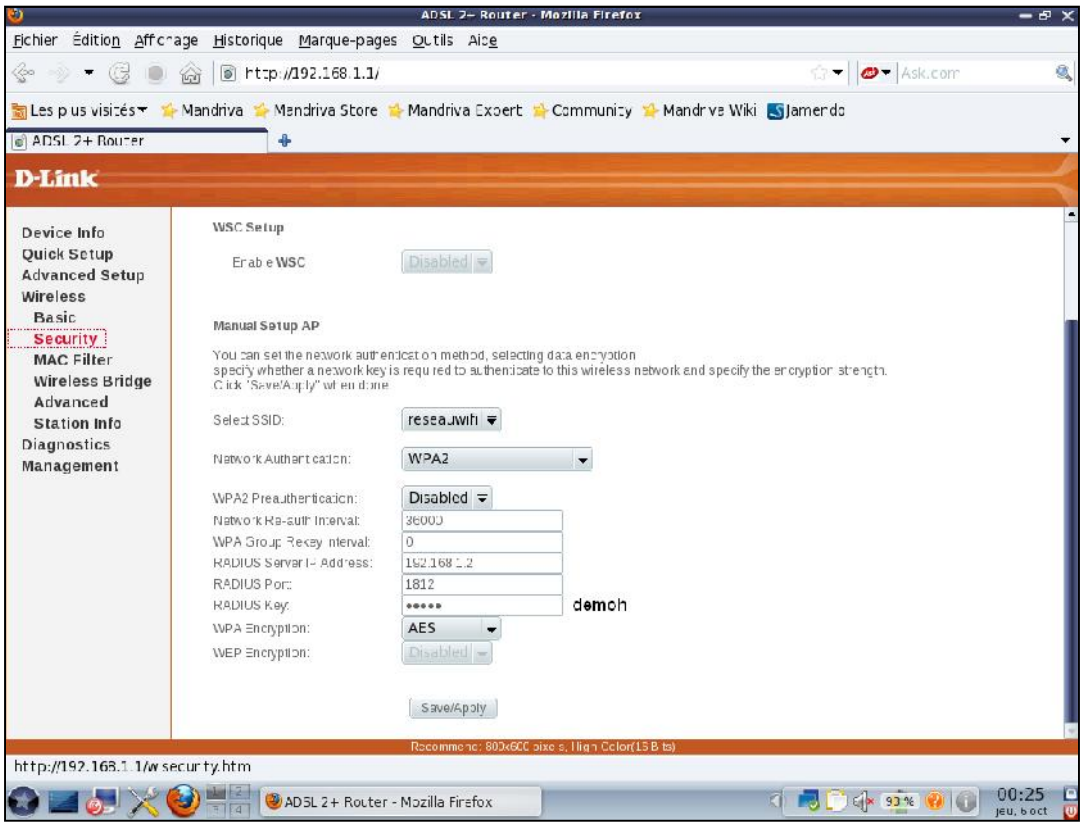

Figure III.12 : Attribution de l'adresse IP du radius à l'AP

Ainsi la configuration est terminée.

5. Configuration du poste client sous Windows XP

La configuration de Windows XP ne doit pas trop poser de problèmes vu qu'il y a tout plein d'assistantes partout.

On dispose déjà des certificats suivants :

5.1. Installation du certificat d'autorité

I l faut simplement double cliquer sur root.der

Etape 1 : cliquer ensuite sur « installer le certificat »

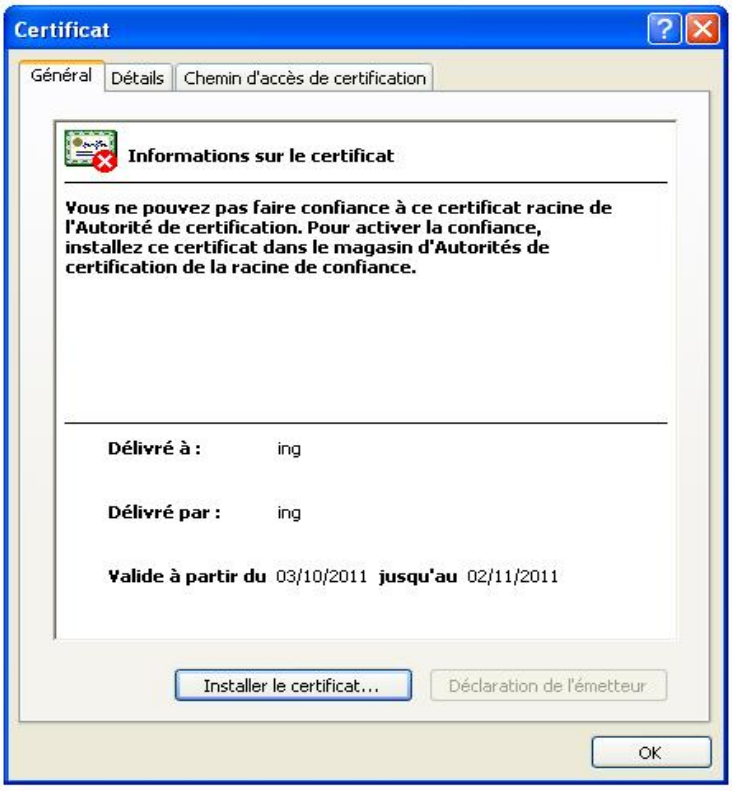

Figure III.13 : Début de l'installation du certificat root

Etape 2 : cliquez sur « suivant »

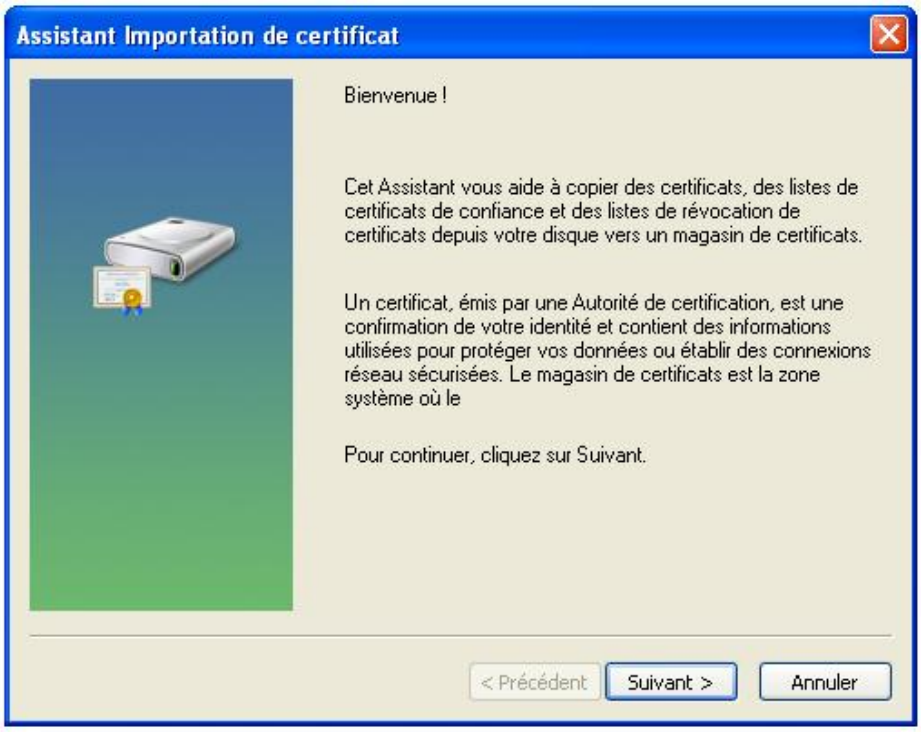

Figure III.14 : Confirmation de l'installation

Etape 3 : cliquez sur « parcourir »

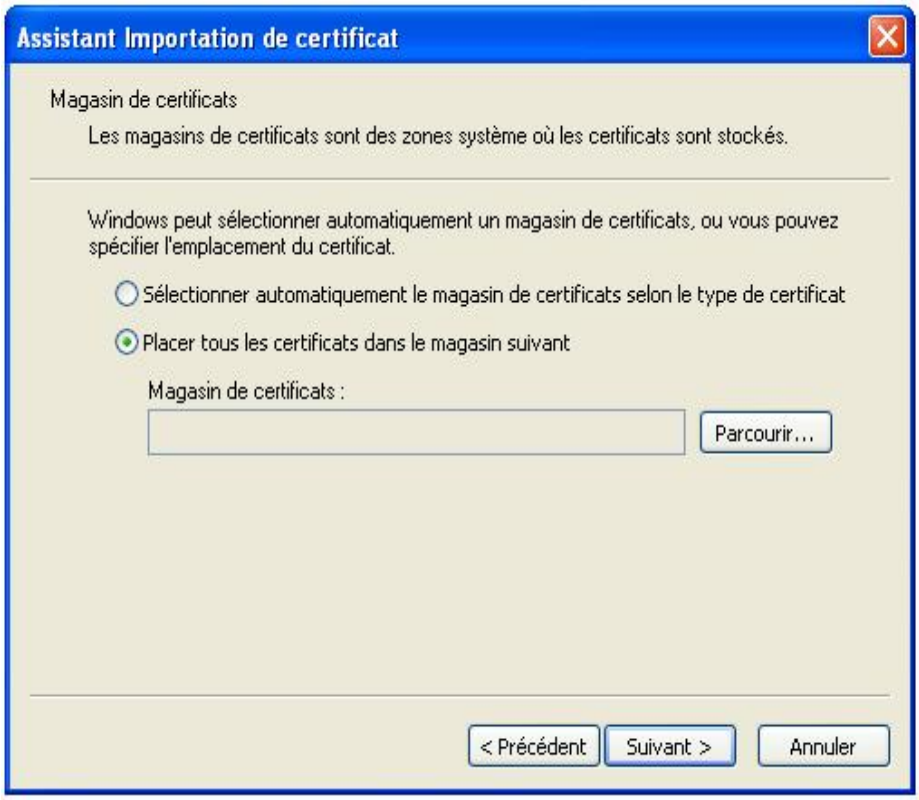

Figure III.15 : Choix de magasin du certificat

Etape 4 : La sélection du magasin de certificat que l'on souhaite utiliser puis « OK »

Figure III.16 : Sélection d'autorités de certification racines de confiance

Etape 5 : « terminer »

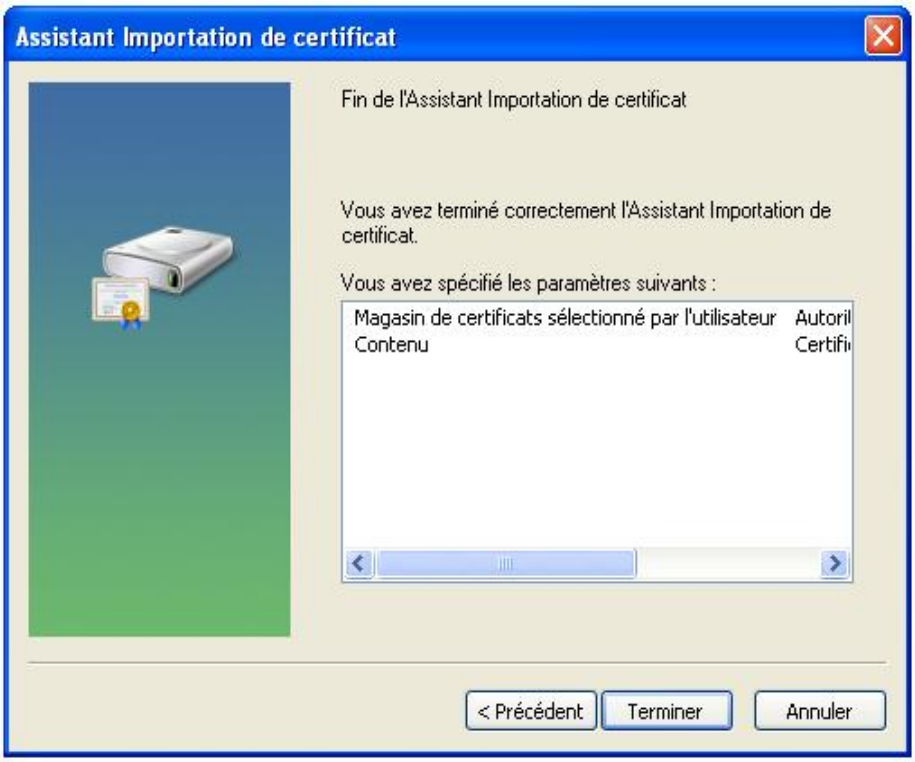

Figure III.17 : Fin de l'importation du certificat

Etape 6 : « oui »

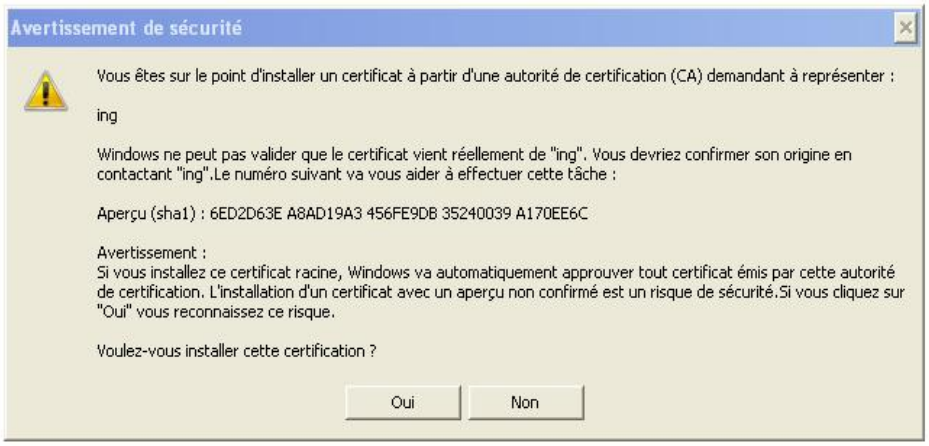

Figure III.18 : Confirmation de la validité du certificat root

Figure III.19 : Fin de l'importation du certificat

L'importation du certificat racine est terminée.

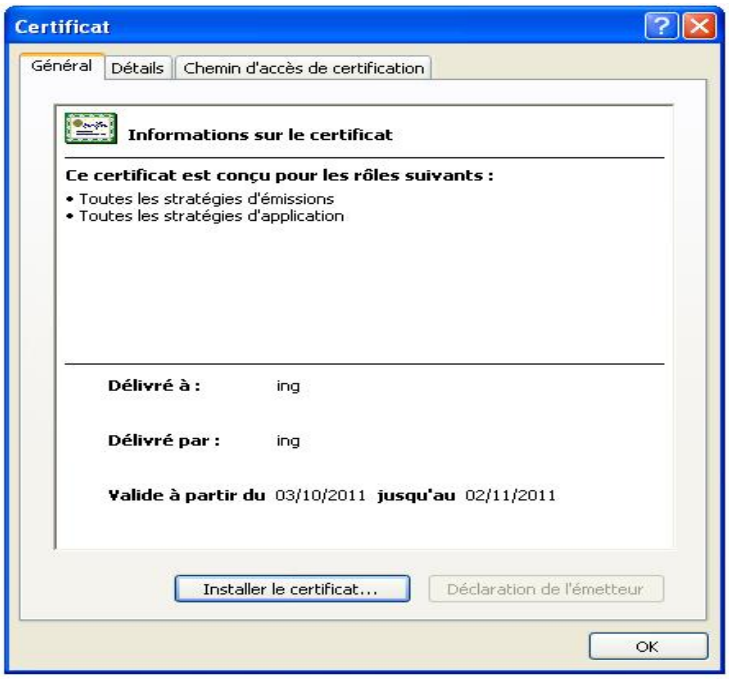

Figure III.20 : Certificat d'autorités root

5.2. Installation du certificat client

Etape 1 : « suivant »

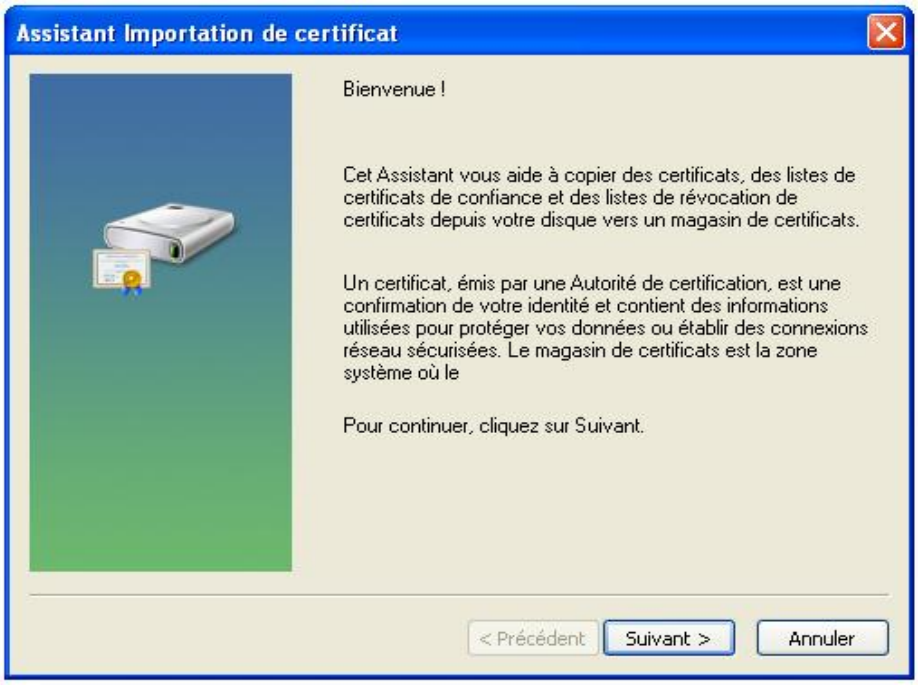

Figure III.21 : Confirmation de l'installation

Etape 2 : « suivant »

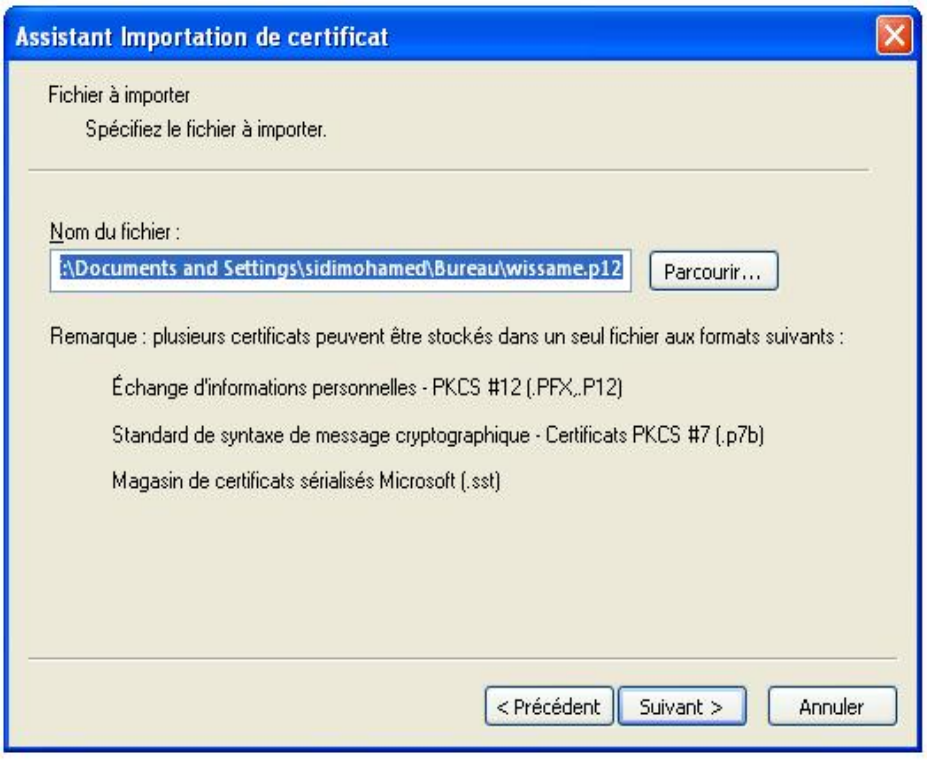

Figure III.22 : Installation du certificat wissame.p12

Etape 3 : On a entré le mot de passe utilisé dans le certificat client (notre mot passe est : whatever)

Figure III.23 : Mot de passe

Etape 4 : sélection du magasin de certificat que l'on veut utilisez puis « OK »

Figure III.24 : Choix de l'emplacement du certificat wissame.p12

Etape 5 : « terminer »

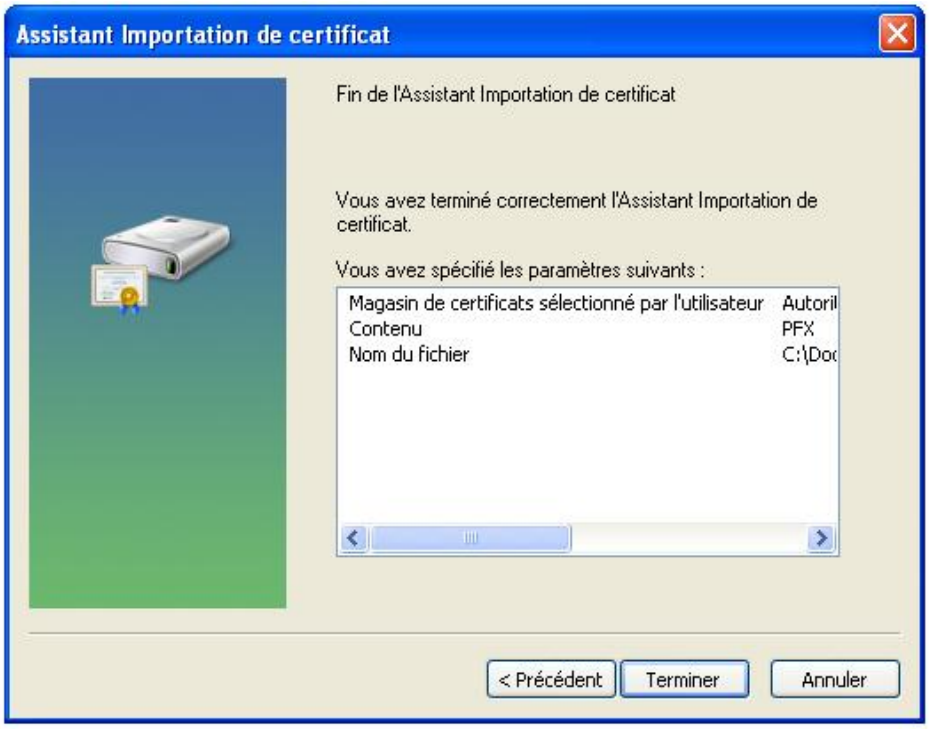

Figure III.25 : Fin de l'importation du certificat

Etape 6 : « OK » pour terminer l'importation

Figure III.26 : Confirmation de la réussite

5.3. Installation du certificat serveur

Identique à celui de certificat client, la seule différence c'est le choix de magasin du certificat qui sera « autorité de certification racine de confiance » et non « personnel ».

6. Configuration de la connexion sans fil

La configuration de la connexion sans fil est très simple.

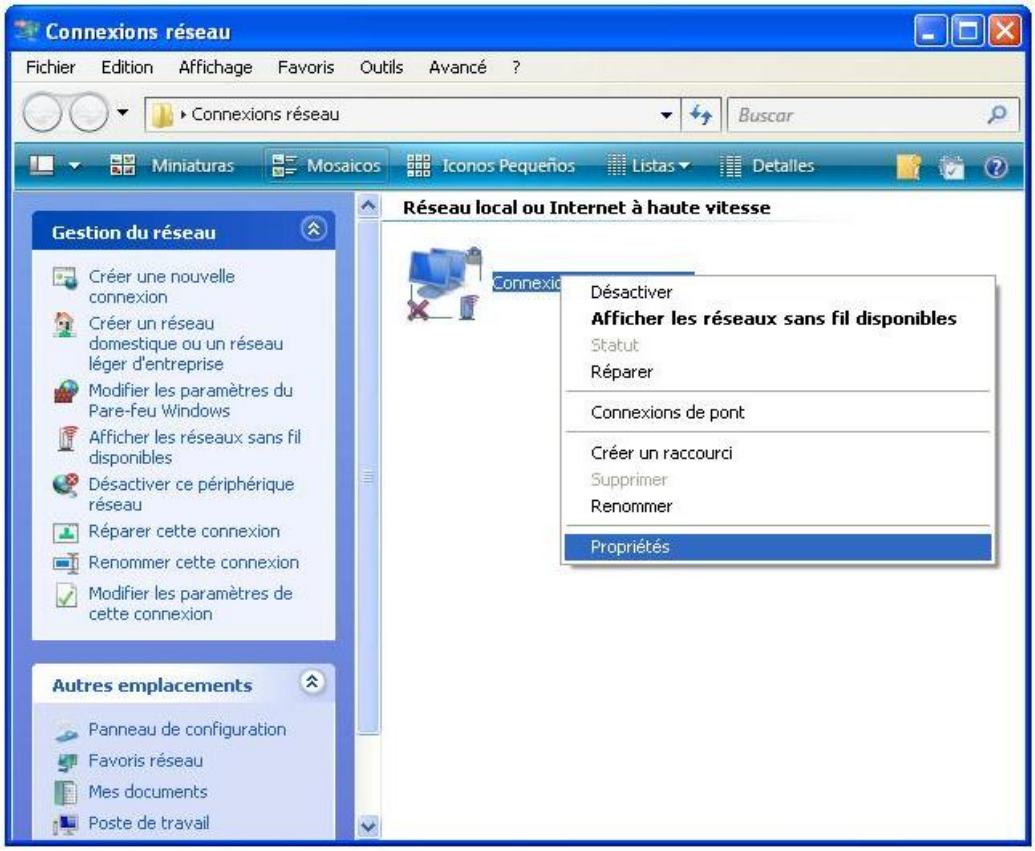

Figure III.27 : Choix de la propriété de la connexion sans fil

Etape 1 : cliquez sur le protocole internet (TCP/IP) puis sur propriétés

Figure III.28 : Choix du Protocole Internet (TCP/IP)

Etape 2 : cliquez sur la configuration réseaux sans fil

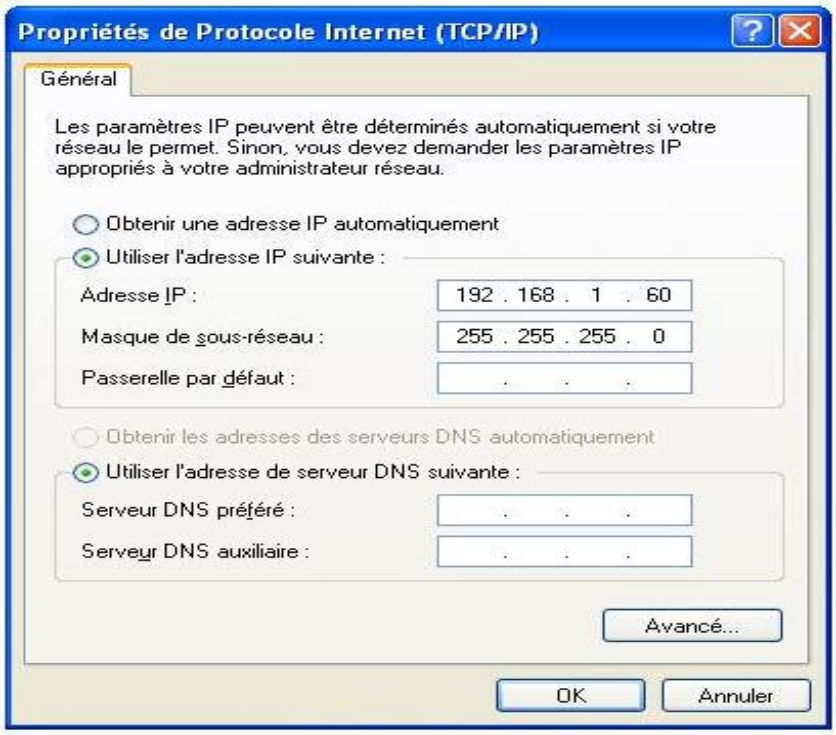

Figure III.29 : Début de la configuration

Etape 3 : sélectionner votre réseau puis cliquez sur propriétés ; s'il n'existe pas dans la liste ajouter-le, ensuite continuer la configuration en cliquant sur propriétés.

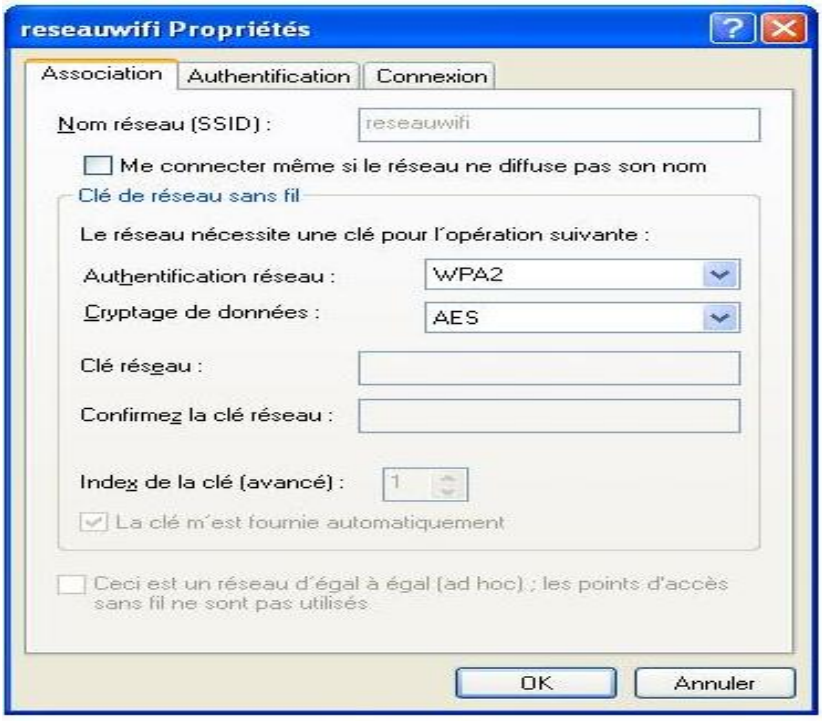

Figure III.30 : Choix de type d'authentification réseau et de type de cryptage des données

Etape 4 : sélectionner le type d'authentification

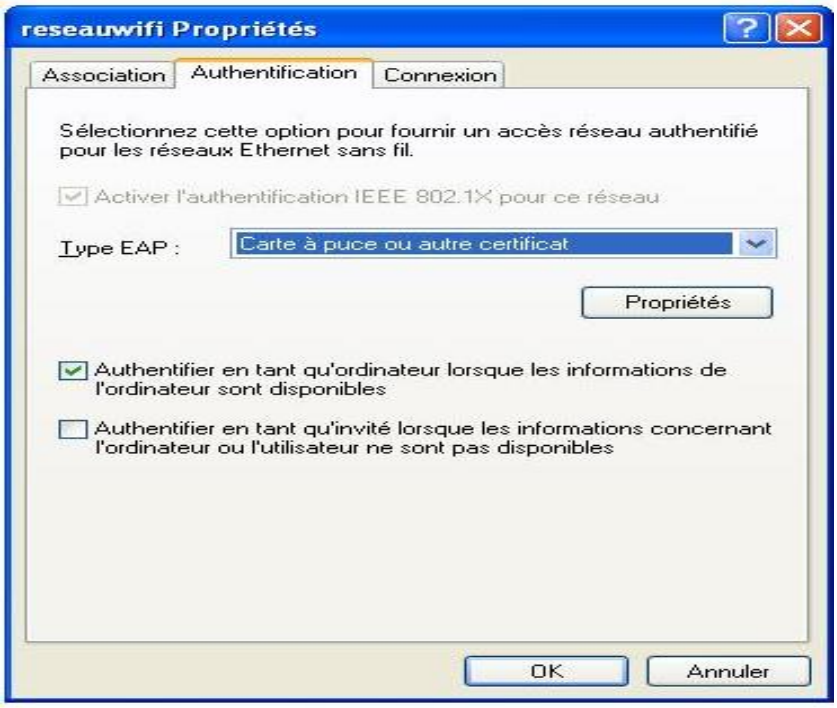

Figure III.31 : Choix du type EAP

Etape 5 : cliquez sur propriété pour choisir les certificats qu'on a installés sur l'ordinateur du client wissame.

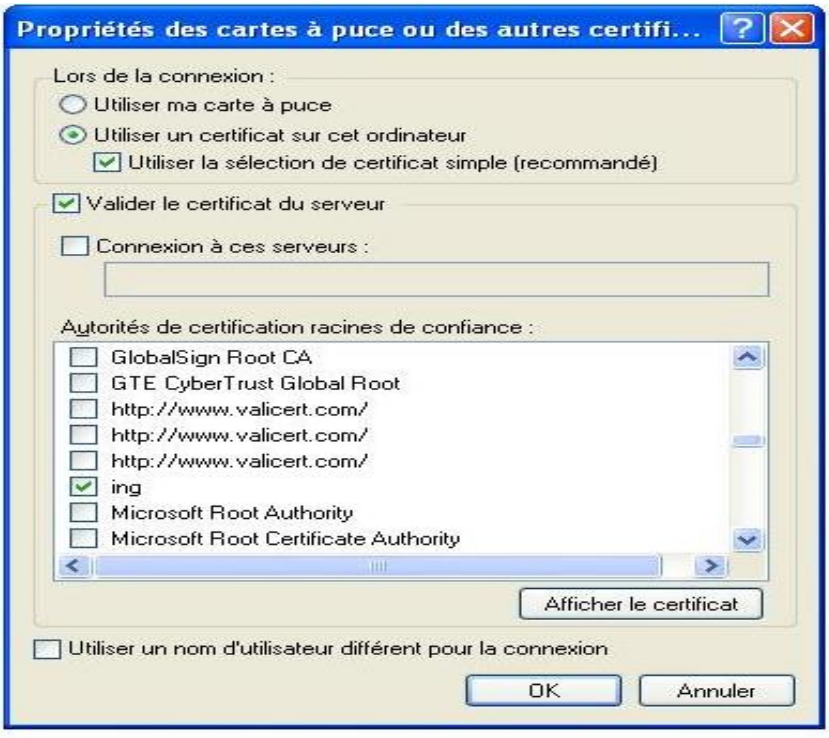

Figure III.32 : Choix du certificat d'autorité root nommé « ing »

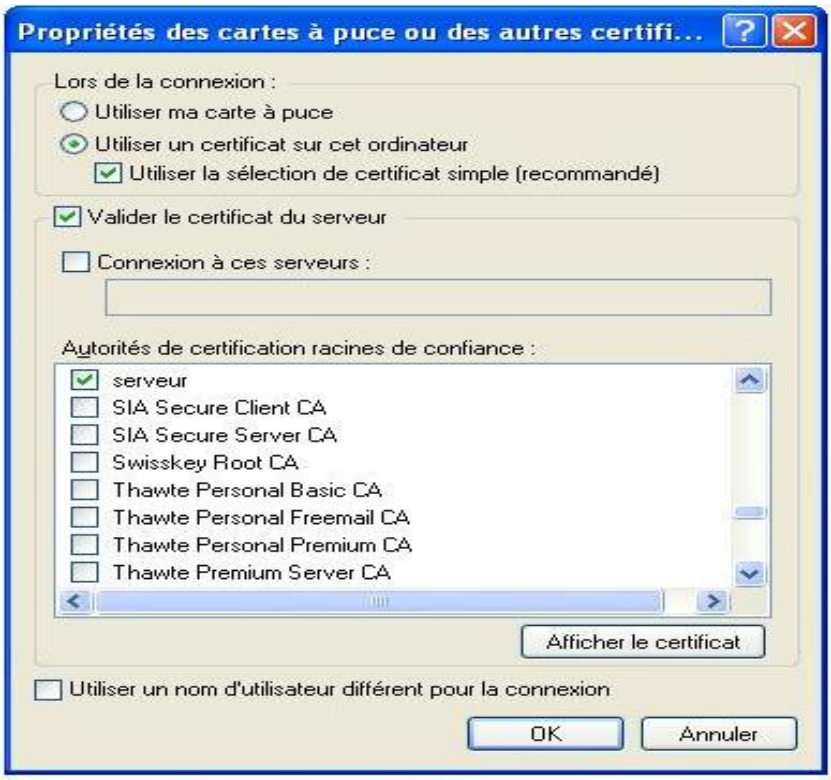

Figure III.33 : Choix du certificat serveur

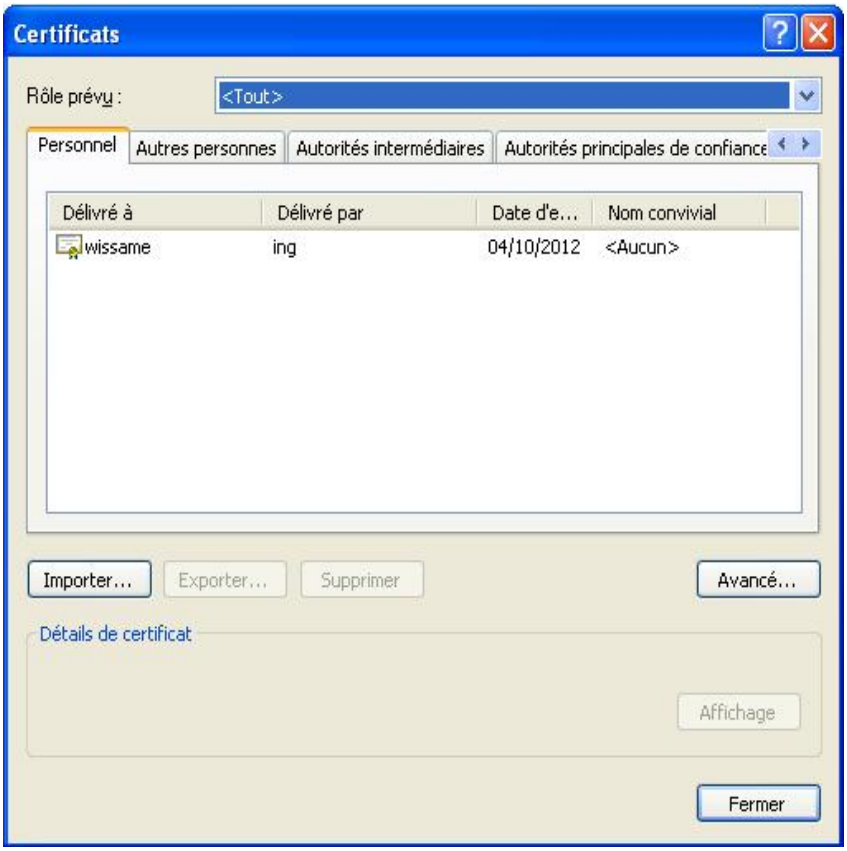

Figure III.34 : Certificat client wissame

Etape 6 : cliquez sur connexion et cochez « Me connecter à ce réseau lorsqu'il est a porté » puis cliquez sur « OK »

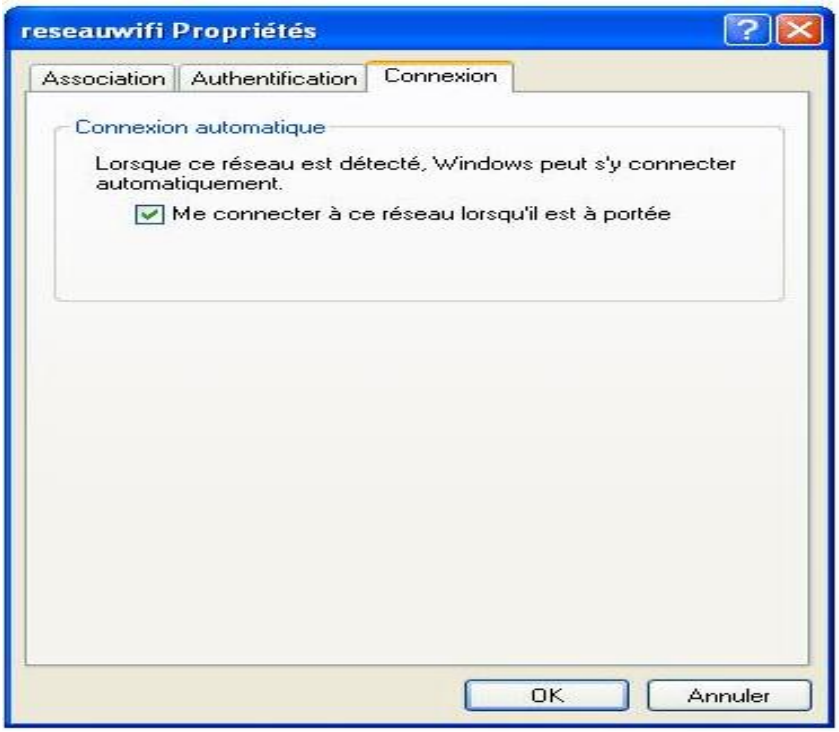

Figure III.35 : Choix de la connexion

Etape 7: cocher « réseaux avec point d'accès seulement (infrastructure) », puis sur fermer

Figure III.36 : Choix du réseau avec point d'accès seulement

Ainsi on a fini la configuration de notre connexion sans fil « reseauwifi ».

Nous allons maintenant essayer de nous connecter au réseau « reseauwifi », mais d'abord on lance le serveur radius.

La carte a détecté un seul réseau sans fil : notre réseau « reseauwifi ».

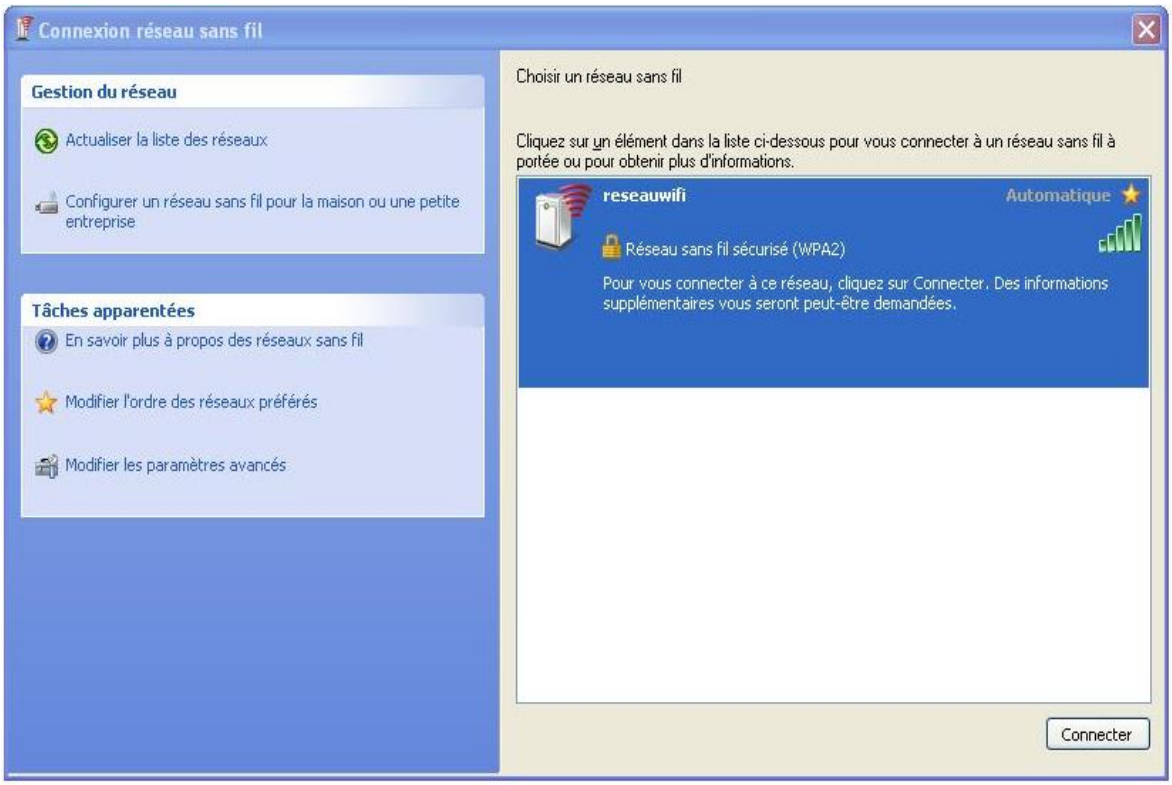

Figure III.37 : Réseau détecté par la carte

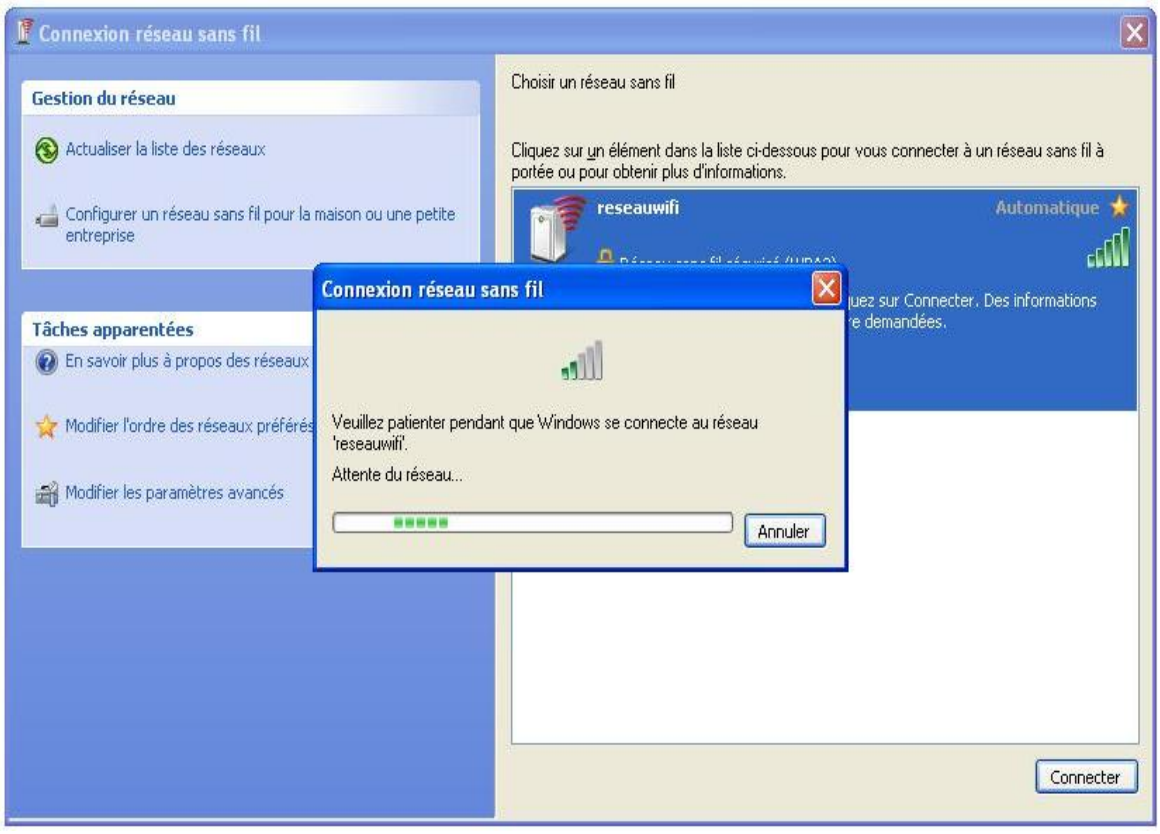

Figure III.38 : Tentative de connexion

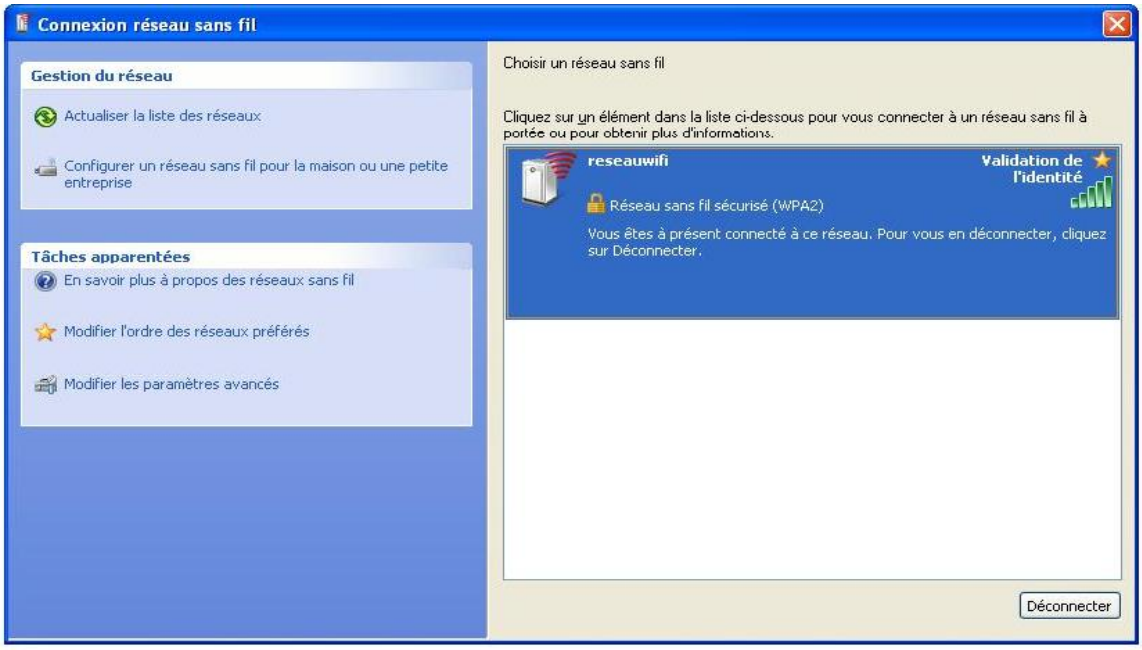

Figure III.39 : Validation de l'identité

Lorsqu'on arrive à s'associer avec le point d'accès et que le radius nous répond :

Radius affiche les échanges des messages EAP entre lui et le client wissame.

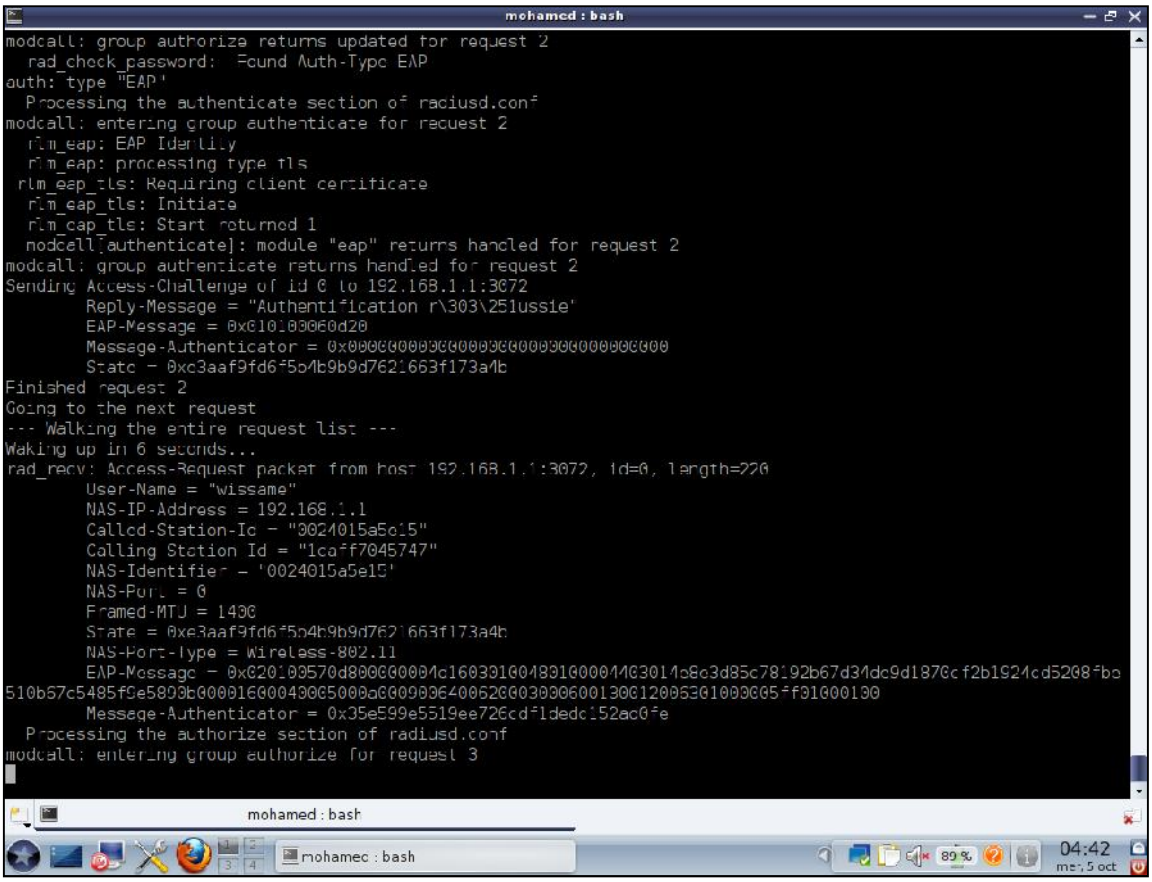

Figure III.40 : Echanges des messages EAP entre le serveur radius et le client wissame

Et en même temps chez le client apparait la fenêtre suivante :

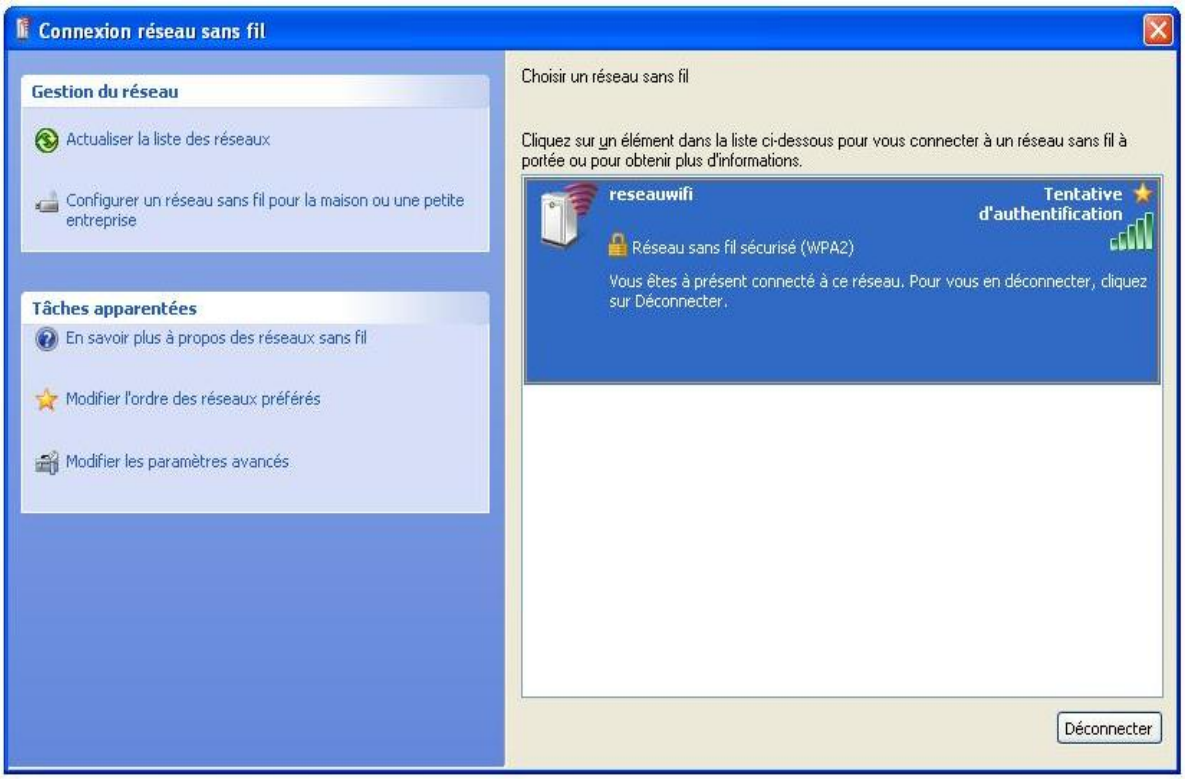

Figure III.41 : Attente de l'authentification

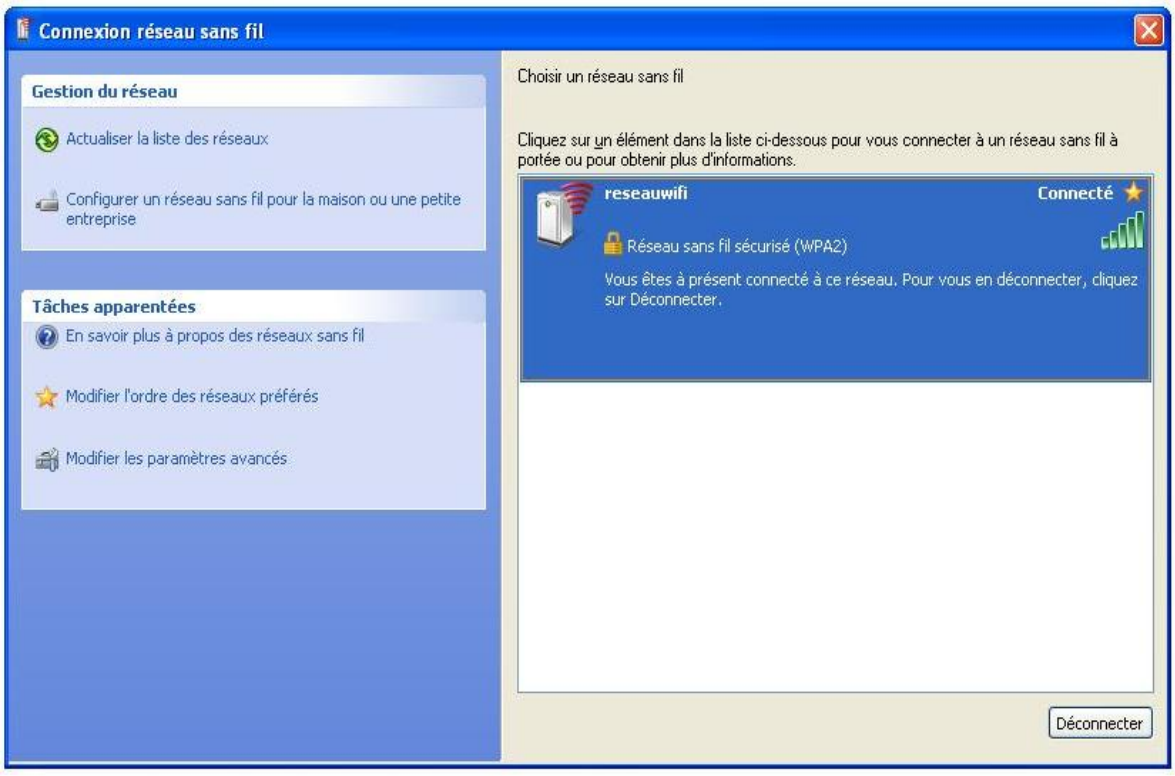

Figure III.42 : Réussite de l'authentification et passage à l'état connecté

Figure III.43 : Etat de connexion réseau sans fil

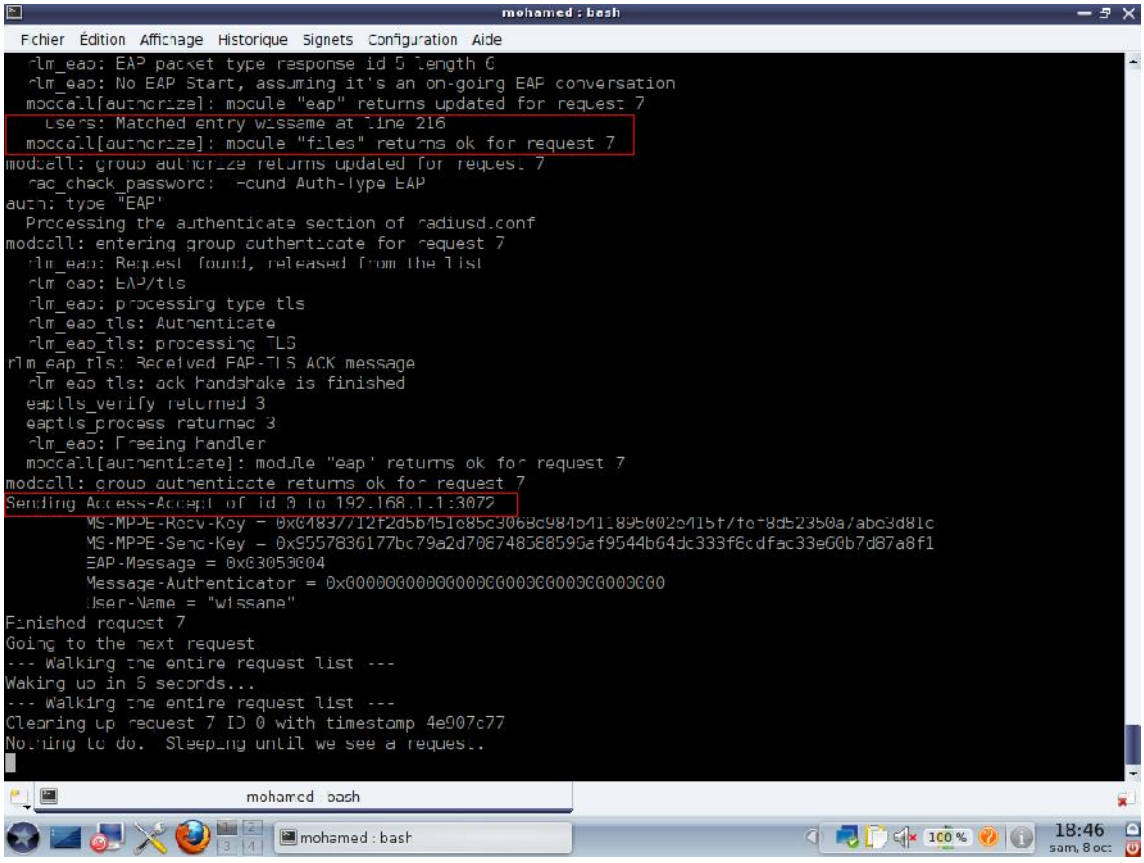

Figure III.44 : Autorisation accepté de serveur pour le client wissame

Pour s'assurer que tout fonctionne bien, on va essayer de faire un partage des fichiers entre le poste serveur et le poste client; (on a trouvé le manuel de configuration de samba pour les partages sous mandriva sur **[23]).**

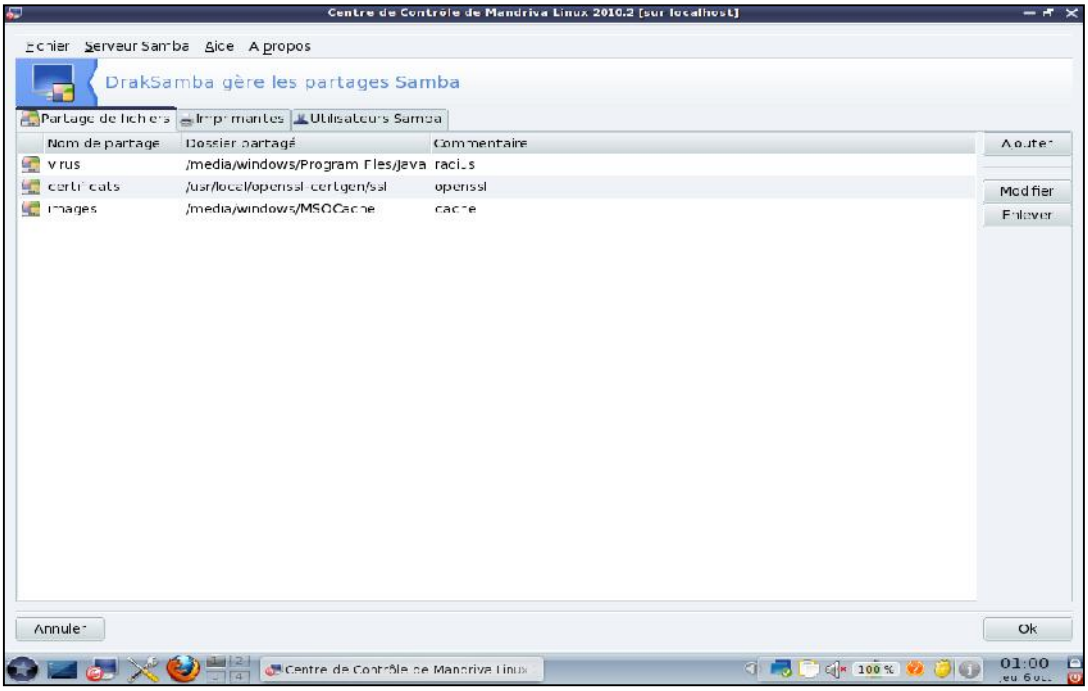

Figure III.45 : Fichiers à partager sur le poste client

Sur le poste client, assurez-vous que le pare-feu autorise les partages des fichiers, cochez « partage de fichiers et d'imprimantes »

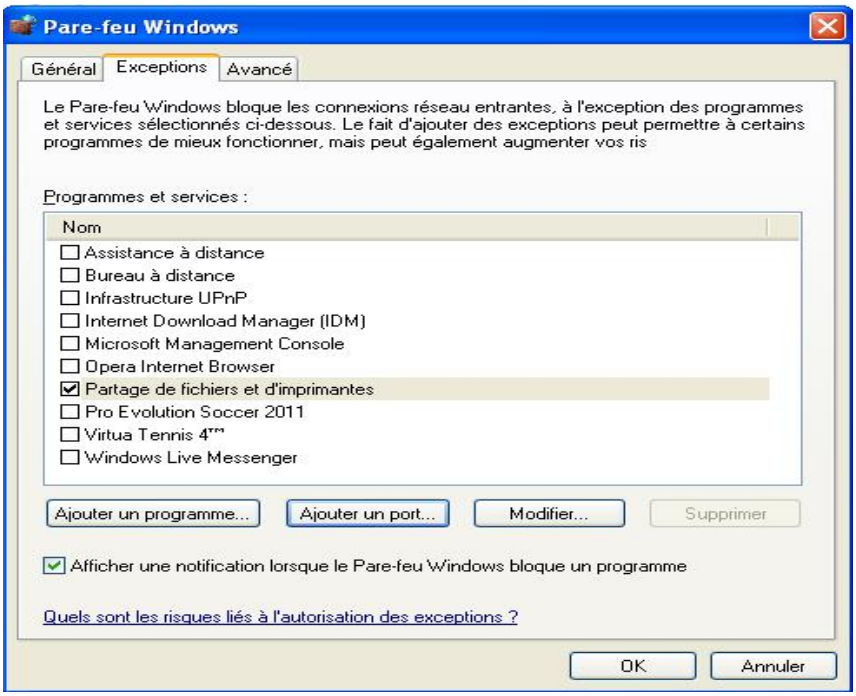

Figure III.46 : Autorisation de partage des fichiers

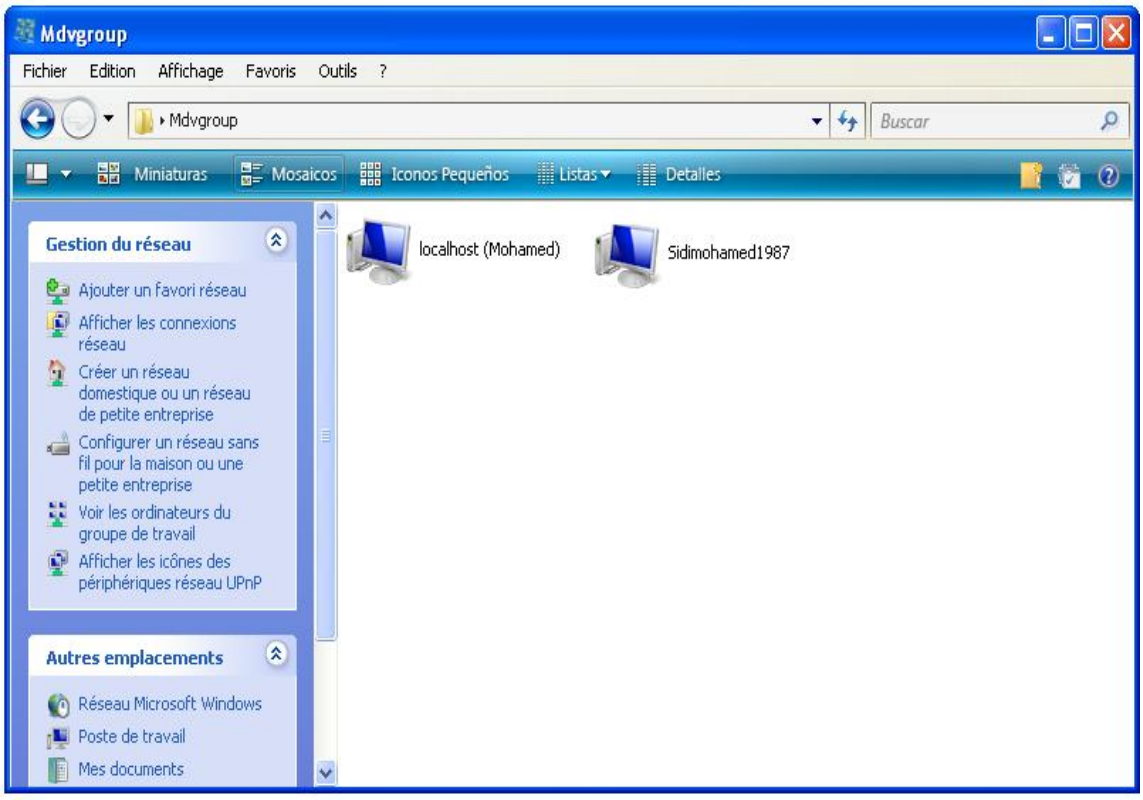

Figure III.47 : Usagers du réseau

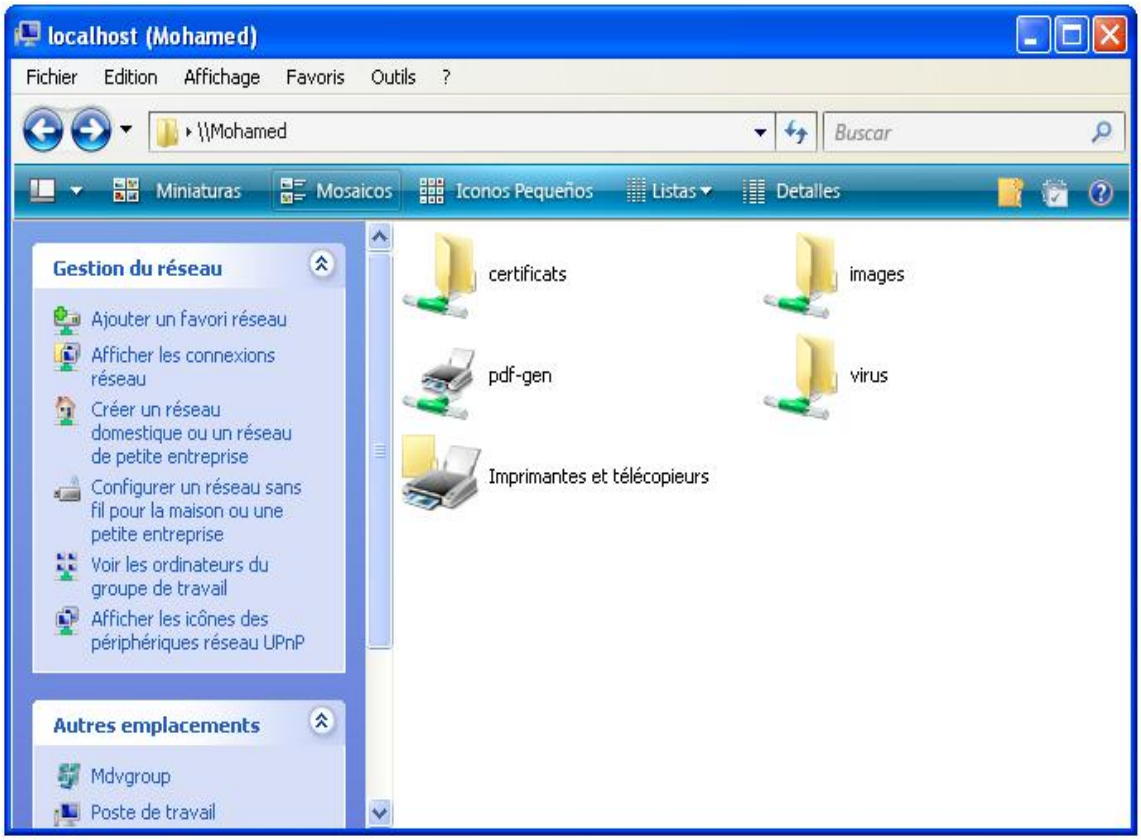

Figure III.48 : Fichiers partagé sur le poste client

Dés qu'il ya d'autres clients connectés sur le réseau, ils peuvent partager des fichiers entre eux.

D'autres solution pour s'assurer de la réussite de connexion c'est d'utilisé le ping entre le serveur et client connecté.

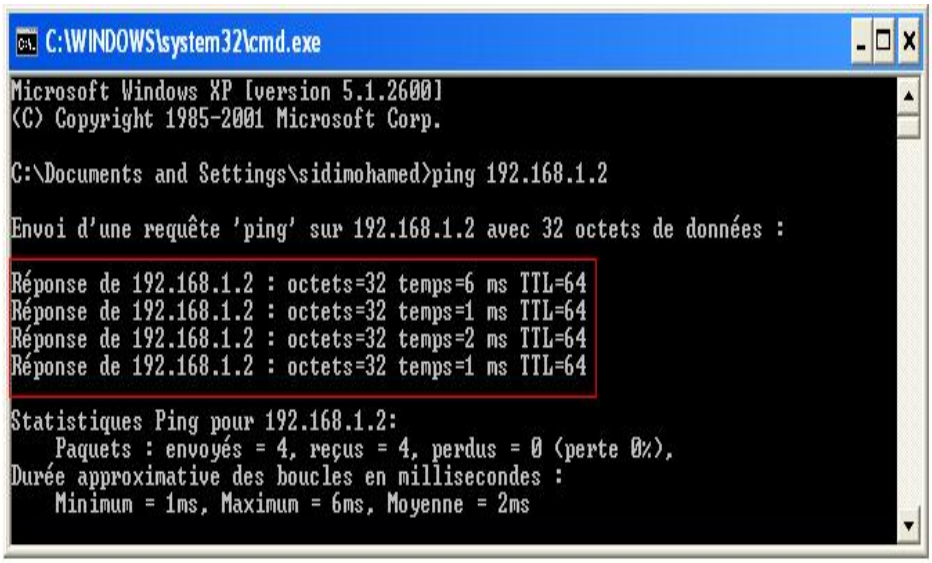

Figure III.49 : Réussite de ping du client vers le serveur

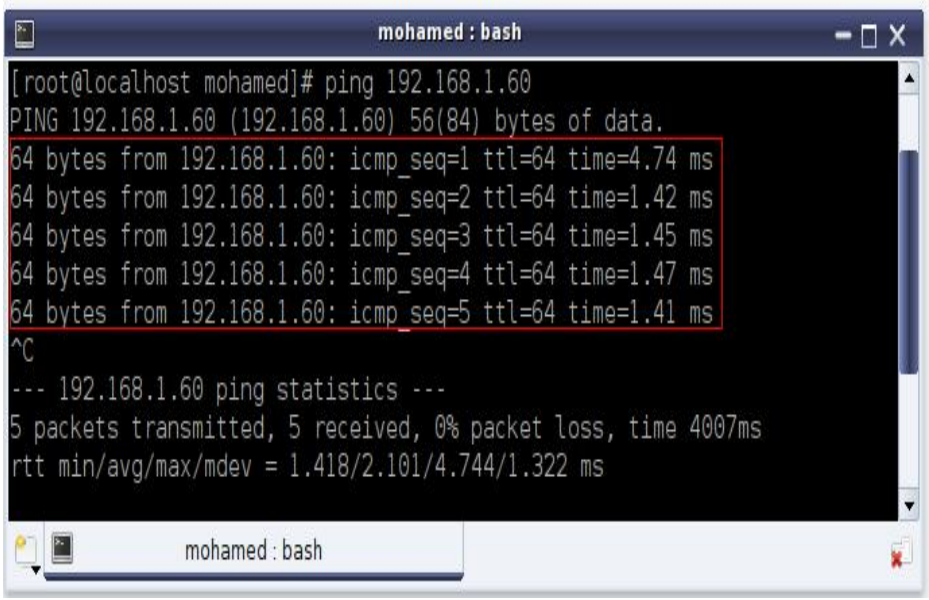

Figure III.50 : Réussite du ping du serveur vers le client

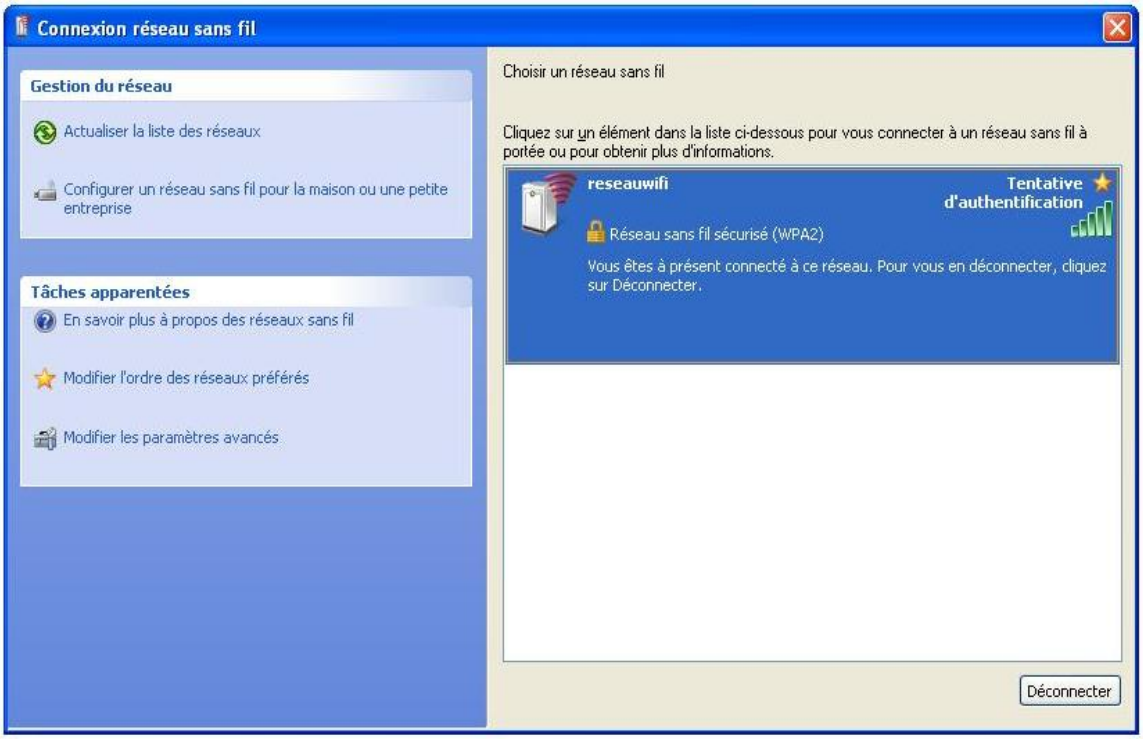

Figure III.51 : Tentative d'un pirate

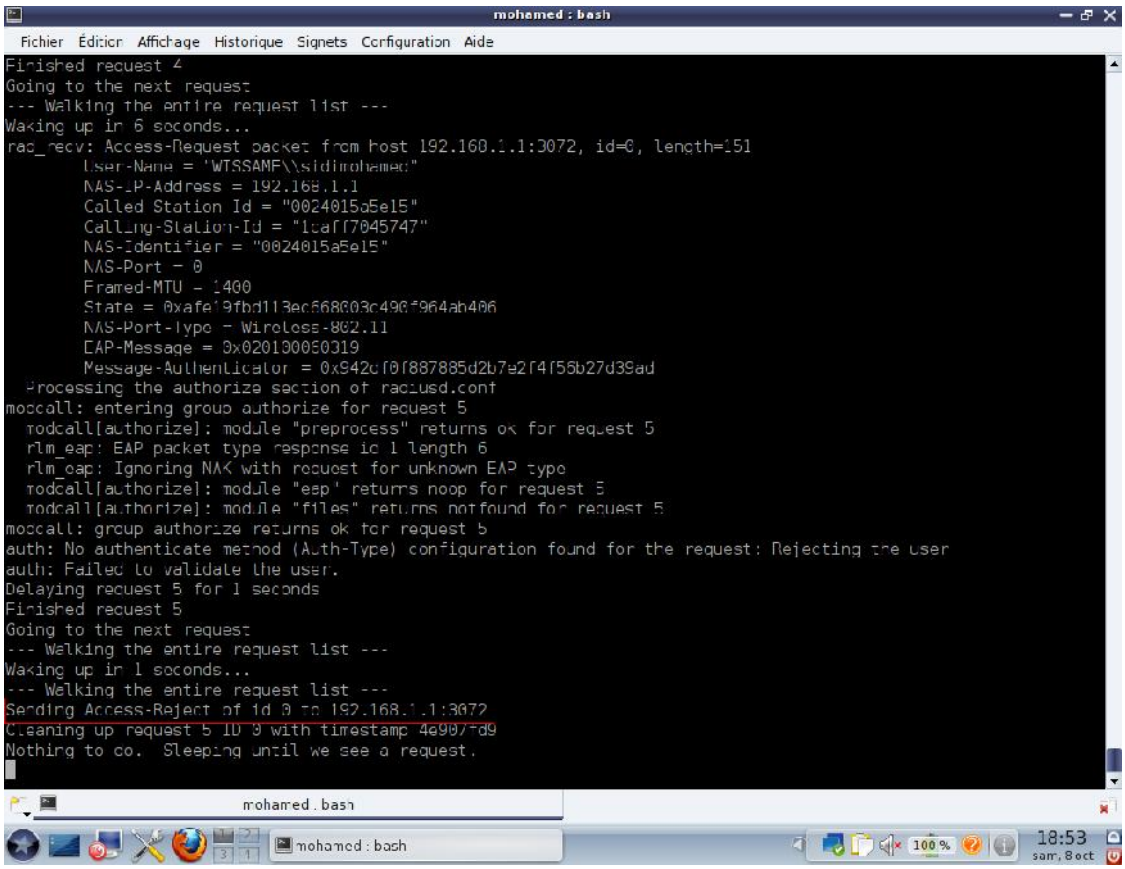

Figure III.52 : Autorisation rejeté du serveur pour le pirate

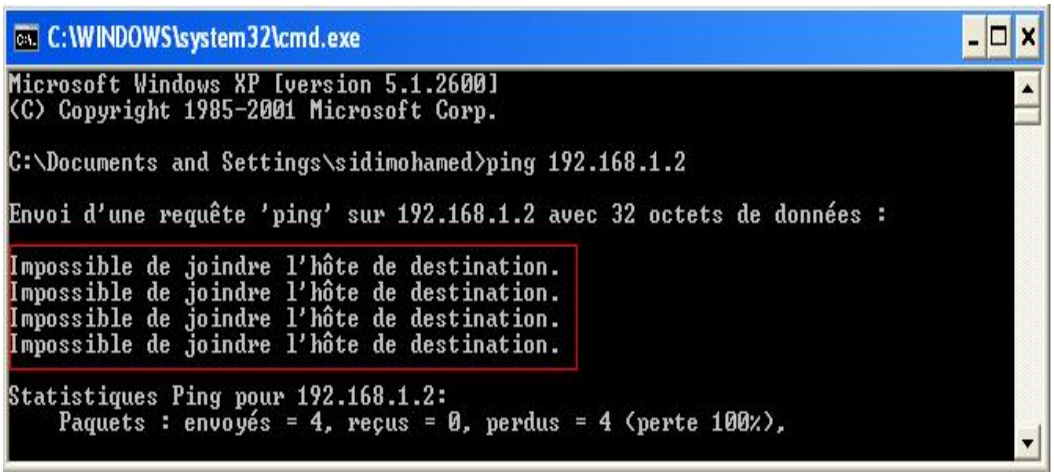

Figure III.53 : Echec du ping

Conclusion

On remarque que le réseau est maintenant sécurisé et que les usagers qui ne sont pas enregistrés dans le serveur comme étant des usagers autorisés, ne pourront pas accéder au réseau, ni même percevoir son existence.

Dans le cas ou la personne est en possession du SSID du réseau, elle ne pourra quand même pas y accéder sans les certificats qui sont installés aussi bien, dans les postes clients que dans le serveur.

L'échange des informations d'authentification, se fait de manière crypté et par un protocole amélioré, qui pour le moment n'a pas été cassé. En plus du fait que le protocole d'authentification 802.1x a donné de très bons résultats pour les réseaux filaires, à travers l'utilisation de serveur d'authentification sous linux, qui comme on le sait ne craint pas les virus.