Hollevoet Gaëtan
Rabas Josselin
Preux Sylvain
Sénécaux Ludovic

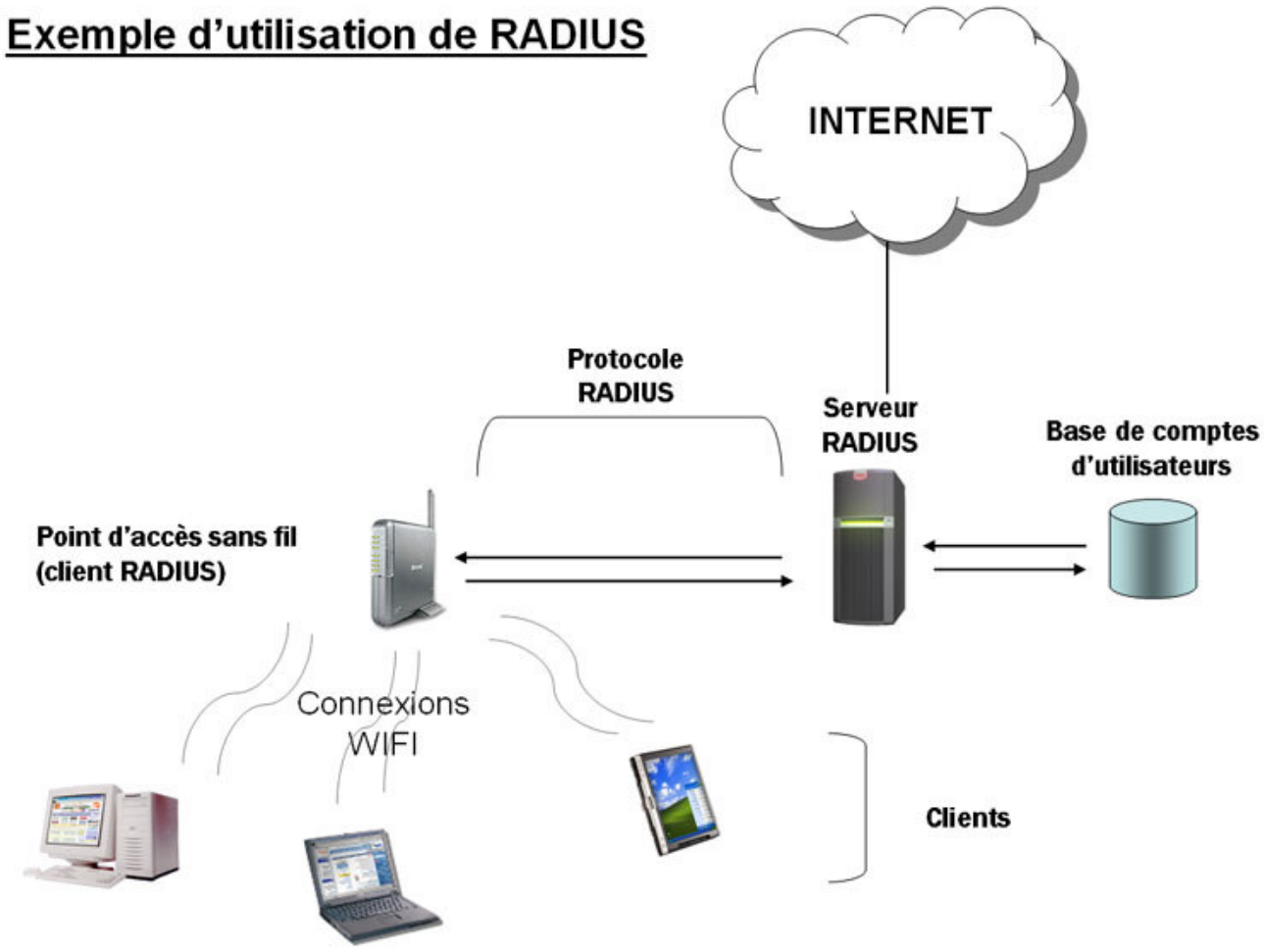
*free*RADIUS

SOMMAIRE

Introduction

1) RADIUS, c'est quoi?

RADIUS (**Remote Authentication Dial-In User Service**) est un protocole client/serveur destiné à permettre à des serveurs d'accès de communiquer avec une base de données centralisée regroupant en un point l'ensemble des utilisateurs distants. Ce serveur central (appelé serveur RADIUS) va authentifier ces utilisateurs, et leur autoriser l'accès à telle ou telle ressource. Une autre fonctionnalité importante d'un serveur RADIUS est la comptabilisation des informations concernant les utilisateurs distants.



2) Matériels et logiciels utilisés

Dans le cadre de notre infrastructure, nous avons utilisé les matériels et logiciels suivants :

- **Matériels :**
  - 1 point d'accès wifi CISCO AIRONET 1200
  - 1 poste serveur
  - 1 poste client avec carte wifi
- **Logiciels :**
  - **Serveur**
    - Linux Mandrake
    - FreeRadius
    - OpenLDAP
    - OpenSSL
  - **Client**
    - Windows XP SP2
    - Linux Mandrake
      - wpa_supplicant

Installation et configuration

A. Côté serveur

1. OpenSSL

OpenSSL (Open Secure Socket Layer) servira à la création des certificats.

- Installation

```
# wget http://www.openssl.org/source/openssl-0.9.7e.tar.gz
...
# tar zxvf openssl-0.9.7e.tar.gz
...
# cd openssl-0.9.7e
# ./config --prefix=/usr/local/openssl-certgen shared
...
# make && make install
...
```

- Configuration

Il faut créer le fichier **xpextensions,** nécessaires à la génération des certificats,
dans le répertoire **/usr/local/openssl-certgen/ssl**

```
[ xpclient_ext]
extendedKeyUsage = 1.3.6.1.5.5.7.3.2
[ xpserver_ext ]
extendedKeyUsage = 1.3.6.1.5.5.7.3.1
```

Création du fichier **CA.root**, shell script de génération du certificat d'autorité :

```
#!/bin/sh
SSL=/usr/local/openssl-certgen
export PATH=${SSL}/bin/:${SSL}/ssl/misc:${PATH}
export LD_LIBRARY_PATH=${SSL}/lib
rm -rf demoCA
echo "**********************************************************************"
echo "Creating self-signed private key and certificate"
echo "When prompted override the default value for the Common Name field"
echo "**********************************************************************"
echo
openssl req -new -x509 -keyout newreq.pem -out newreq.pem -passin pass:whatever
  -passout pass:whatever
echo "**********************************************************************"
echo "Creating a new CA hierarchy (used later by the "ca" command) with the
  certificate"
echo "and private key created in the last step"
echo "**********************************************************************"
```

```
echo
echo "newreq.pem" | CA.pl -newca >/dev/null
echo "**********************************************************************"
echo "Creating ROOT CA"
echo "**********************************************************************"
openssl pkcs12 -export -in demoCA/cacert.pem -inkey newreq.pem -out root.p12 -cacerts
  -passin pass:whatever -passout pass:whatever
openssl pkcs12 -in root.p12 -out root.pem -passin pass:whatever -passout pass:whatever
openssl x509 -inform PEM -outform DER -in root.pem -out root.der
rm -rf newreq.pem
```

Création du fichier **CA.svr,** shell script de génération du certificat serveur :

```
#!/bin/sh
SSL=/usr/local/openssl-certgen
export PATH=${SSL}/bin/:${SSL}/ssl/misc:${PATH}
export LD_LIBRARY_PATH=${SSL}/lib
echo "**********************************************************************"
echo "Creating server private key and certificate"
echo "When prompted enter the server name in the Common Name field."
echo "**********************************************************************"
openssl req -new -keyout newreq.pem -out newreq.pem -passin pass:whatever
  -passout pass:whatever
openssl ca -policy policy_anything -out newcert.pem -passin pass:whatever
  -key whatever -extensions xpserver_ext -extfile xpextensions -infiles newreq.pem
openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -out $1.p12 -clcerts
  -passin pass:whatever -passout pass:whatever
openssl pkcs12 -in $1.p12 -out $1.pem -passin pass:whatever -passout pass:whatever
openssl x509 -inform PEM -outform DER -in $1.pem -out $1.der
rm -rf newert.pem newreq.pem
```

Nous pouvons désormais passé à la génération des certificats.

a) Génération du certificat d'autorité

```
# cd /usr/local/openssl-certgen/ssl
# ./CA.root
...
```

b) Génération du certificat serveur

Attention ! Mettez bien dans le **CommonName** le nom de votre serveur passé en paramètre

```
# ./CA.svr server-radius1.univ-lille1.fr
...
 Common Name (eg, YOUR name) []:server-radius1.univ-lille1.fr
...
 A challenge password []:whatever
...
 Sign the certificate? [y/n]:y
...
 1 out of 1 certificate requests certified, commit? [y/n]y
...
```

c) Installation des certificats et compilation du programme **random**

- Installation des certificats

Nous allons un peu anticiper l'installation de FreeRADIUS en créant son répertoire de configuration :

```
# mkdir -p /etc/raddb
# mkdir -p /etc/raddb/certs
# cp root.* server-radius1.univ-lille1.fr.* /etc/raddb/certs/
# cd /etc/raddb/certs
```

Création du fichier **dh** :

```
# openssl dhparam -check -test -5 512 -out dh
...
```

- Programme **random**

Il faut installer les librairies **SSL** :

```
# urpmi libopenssl-devel
```

Ensuite, nous pouvons créer le fichier source **random.c** :

```
#include <stdio.h>
#include <openssl/rand.h>

main(void)
{
   unsigned char buf[100];
   if (!RAND_bytes(buf, 100)) {
   }
   printf("Random : %s\n", buf);
}
```

Et le compiler :

```
# gcc random.c -o random -lcrypto
```

2. OpenLDAP

OpenLDAP (Open Lightweight Directory Access Protocol) est un annuaire qui contiendra les comptes utilisateurs.

- Installation

```
# urpmi openldap openldap-clients openldap-servers libldap2-devel
...
```

- Configuration

Modification du fichier **/etc/openldap/slapd.conf** :

```
# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.8.8.6 2001/04/20 23:32:43 kurt Exp

include     /usr/share/openldap/schema/core.schema
include     /usr/share/openldap/schema/cosine.schema
include     /usr/share/openldap/schema/corba.schema
include     /usr/share/openldap/schema/inetorgperson.schema
include     /usr/share/openldap/schema/java.schema
include     /usr/share/openldap/schema/krb5-kdc.schema
include     /usr/share/openldap/schema/kerberosobject.schema
include     /usr/share/openldap/schema/misc.schema
include     /usr/share/openldap/schema/nis.schema
include     /usr/share/openldap/schema/openldap.schema
include     /usr/share/openldap/schema/autofs.schema
include     /usr/share/openldap/schema/samba.schema
include     /usr/share/openldap/schema/kolab.schema
include     /usr/share/openldap/schema/evolutionperson.schema
include     /usr/share/openldap/schema/calendar.schema
include     /usr/share/openldap/schema/sudo.schema
include     /usr/share/openldap/schema/dnszone.schema
include     /usr/share/openldap/schema/dhcp.schema
include     /etc/openldap/schema/local.schema
include     /etc/openldap/slapd.access.conf

pidfile     /var/run/ldap/slapd.pid
argsfile    /var/run/ldap/slapd.args
modulepath  /usr/lib/openldap

# To allow TLS-enabled connections, create /etc/ssl/openldap/ldap.pem
# and uncomment the following lines.
#TLSRandFile            /dev/random
#TLSCipherSuite         HIGH:MEDIUM:+SSLv2
TLSCertificateFile      /etc/ssl/openldap/ldap.pem
TLSCertificateKeyFile   /etc/ssl/openldap/ldap.pem
#TLSCACertificatePath   /etc/ssl/openldap/
#TLSCACertificateFile    /etc/ssl/cacert.pem
TLSCACertificateFile    /etc/ssl/openldap/ldap.pem
#TLSVerifyClient never # ([never]|allow|try|demand)

loglevel 256

schemacheck off
########################################################################
# database definitions
########################################################################
```

```
database    bdb
suffix      "dc=univ-lille1,dc=fr"
rootdn      "cn=admin,dc=univ-lille1,dc=fr"
rootpw      {SSHA}d4VNtKAPKiCXQNrYkd+0663BRAUN/j2L

directory   /var/lib/ldap

checkpoint 256 5

index objectClass,uid,uidNumber,gidNumber,memberuid  eq
index cn,mail,surname,givenname           eq,subinitial
```

Le mot de passe crypté (**rootpw**) s'obtient par la commande suivante :

```
# slappasswd -h {SSHA} -s admin
...
```

Modification du fichier **/etc/openldap/slapd.access.conf** :

```
access to *
      by dn="cn=admin,dc=univ-lille1,dc=fr" write
      by dn="uid=radius,ou=utilisateurs,ou=radius,dc=univ-lille1,dc=fr" read
      by anonymous auth
      by self write
      by * none
```

Création du fichier **base.ldif** qui contiendra les informations nécessaires au fonctionnement de l'annuaire :

```
dn: dc=univ-lille1,dc=fr
objectClass: dcObject
objectClass: organization
dc: univ-lille1
o: univ-lille1

dn: cn=admin,dc=univ-lille1,dc=fr
objectClass: organizationalRole
cn: admin

dn: ou=radius,dc=univ-lille1,dc=fr
objectClass: organizationalUnit
ou: radius

dn: ou=utilisateurs,ou=radius,dc=univ-lille1,dc=fr
objectClass: organizationalUnit
ou: utilisateurs

dn: ou=groupes,ou=radius,dc=univ-lille1,dc=fr
objectClass: organizationalUnit
ou: groupes
```

Entrer les informations dans l'annuaire :

```
# ldapadd -x -h 127.0.0.1 -D "cn=admin,dc=univ-lille1,dc=fr" -W -f base.ldif
...
```

3. FreeRadius

FreeRadius (Free Remote Authentication Dial-In User Service) servira à authentifier les clients.

- Installation

```
# urpmi freeradius libfreeradius1-ldap
...
```

- Configuration

Modification du fichier **/etc/raddb/clients.conf** :

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

client 192.168.0.250 {
   secret = cisco
   shortname = cisco_aironet_1200
   nastype = cisco
}
```

Modification du fichier **/etc/raddb/eap.conf** :

```
eap {
   default_eap_type = peap
   timer_expire = 60
   ignore_unknown_eap_types = yes
   cisco_accounting_username_bug = no

   tls {
      private_key_password = whatever
      private_key_file = ${raddbdir}/certs/server-radius1.univ-lille1.fr.pem
      certificate_file = ${raddbdir}/certs/server-radius1.univ-lille1.fr.pem
      CA_file = ${raddbdir}/certs/root.pem
      CA_path = ${raddbdir}/certs/
      dh_file = ${raddbdir}/certs/dh
      random_file = ${raddbdir}/certs/random
      fragment_size = 1024
      include_lenght = yes
      check_crl = yes
   }

   peap {
      default-eap-type = mschapv2
   }

   mschapv2 {
   }
}
```

Modification du fichier **/etc/raddb/users** :

```
DEFAULT Ldap-Group == desactive,
        User-Profile := "cn=desactive,ou=groupes=ou=radius,dc=univ-lille1,dc=fr"
        Reply-Message = "compte desactive"

DEFAULT Ldap-Group == etudiant,
        User-Profile := "cn=etudiant,ou=groupes=ou=radius,dc=univ-lille1,dc=fr"
        Fall-Through = no

DEFAULT Ldap-Group == personnel,
        User-Profile := "cn=personnel,ou=groupes=ou=radius,dc=univ-lille1,dc=fr"
        Fall-Through = no

DEFAULT Ldap-Group == admin,
        User-Profile := "cn=admin,ou=groupes=ou=radius,dc=univ-lille1,dc=fr"
        Fall-Through = no

DEFAULT Auth-Type := Reject
        Reply-Message = "acces interdit"
```

Modification du fichier **/etc/raddb/radiusd.conf** :

```
prefix = /usr
exec_prefix = ${prefix}
sysconfdir = /etc
localstatedir = /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/freeradius
pidfile = ${run_dir}/radiusd.pid
...
user = radius
group = radius
...
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
   ...
```

```
   ldap {
      server = "localhost"
      identity = "uid=radius,ou=utilisateurs,ou=radius,dc=univ-lille1,dc=fr"
      password = radius
      basedn = "ou=radius,dc=univ-lille1,dc=fr"
      filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
      dictionary_mapping = ${raddbdir}/ldap.attrmap
      start_tls = no
      password_attribute = userPassword
      groupname_attribute = radiusGroupName
      groupmembership_filter =
         "(&(uid={Stripped-User-Name:-%{User- Name}}(objectclass=radiusprofile))"
      groupmembership_attribute = radiusGroupName
   }
   ...
   detail auth_log {
      detailfile = ${radacctdir}/%{Client-IP-Address}/auth-detail-%Y%m%d
      detailperm = 0600
   }
   ...
}
authorize {
   preprocess
   auth_log
   eap
   files
   ldap
}
...
authenticate {
   unix
   Auth-Type LDAP {
      ldap
   }
   Auth-Type EAP {
      eap
   }
}
...
```

Configuration des **CRL** (Certificate Revocation List), qui permettent de vérifier si les certificats utilisés sont toujours valides ou non :

```
# cp /usr/local/openssl-certgen/ssl/demoCA/index.txt /etc/raddb/certs/
# c_rehash /etc/raddb/certs/
...
```

Afin que FreeRadius communique avec OpenLDAP, il faut modifier le fichier de configuration de l'annuaire :

```
...
include /usr/share/doc/freeradius-1.0.0/doc/RADIUS-LDAPv3.schema
...
```

On redémarre le serveur LDAP :

```
# /etc/init.d/ldap restart
```

Il faut ajouter l'utilisateur **radius**, ainsi que les groupes, dans l'annuaire afin d'établir la connexion.
Création du fichier **radius.ldif** qui contient l'entrée de cet utilisateur et des groupes :

```
dn: cn=admin,ou=groupes,ou=radius,dc=univ-lille1,dc=fr
objectClass: radiusprofile
radiusAuthType: EAP
radiusServiceType: Framed-User
cn: admin

dn: uid=radius,ou=utilisateurs,ou=radius,dc=univ-lille1,dc=fr
objectClass: radiusprofile
uid: radius
userPassword: radius
radiusGroupName: admin

dn: cn=desactive,ou=groupes,ou=radius,dc=univ-lille1,dc=fr
objectClass: radiusprofile
radiusAuthType: Reject
cn: desactive

dn: cn=etudiant,ou=groupes,ou=radius,dc=univ-lille1,dc=fr
objectClass: radiusprofile
radiusAuthType: EAP
radiusServiceType: Framed-User
cn: etudiant

dn: cn=personnel,ou=groupes,ou=radius,dc=univ-lille1,dc=fr
objectClass: radiusprofile
radiusAuthType: EAP
radiusServiceType: Framed-User
cn: personnel
```

On ajoute ces entrées dans l'annuaire :

```
# ldapadd -x -h 127.0.0.1 -D "cn=admin,dc=univ-lille1,dc=fr" -W -f radius.ldif
...
```

On ajoute les directives suivantes au fichier **/etc/raddb/ldap.attrmap** :

```
...
checkItem          Auth-Type          radiusAuthType
checkItem          Ldap-Group         radiusGroupName
...
```

Nous pouvons désormais démarrer le serveur RADIUS :

```
# radiusd -X -A &
  ...
  ...
  ...
  Listening on authentication *:1812
  Listening on accounting *:1813
  Ready to process requests.
```

B. Côté NAS – Network Access Server

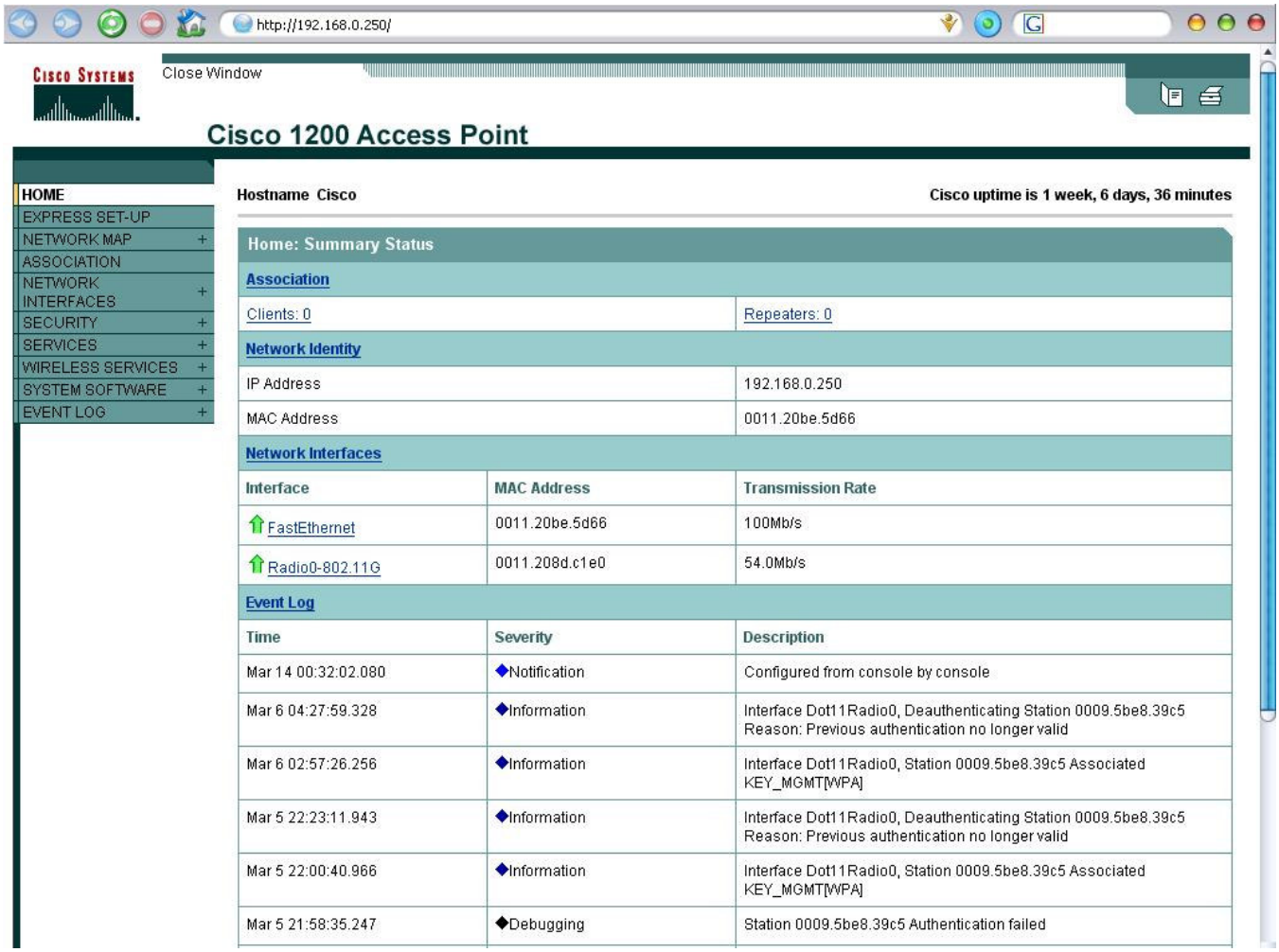
Le **NAS** est le matériel qui va recevoir la demande de connexion d'un client.
Ici, c'est notre point d'accès wifi.

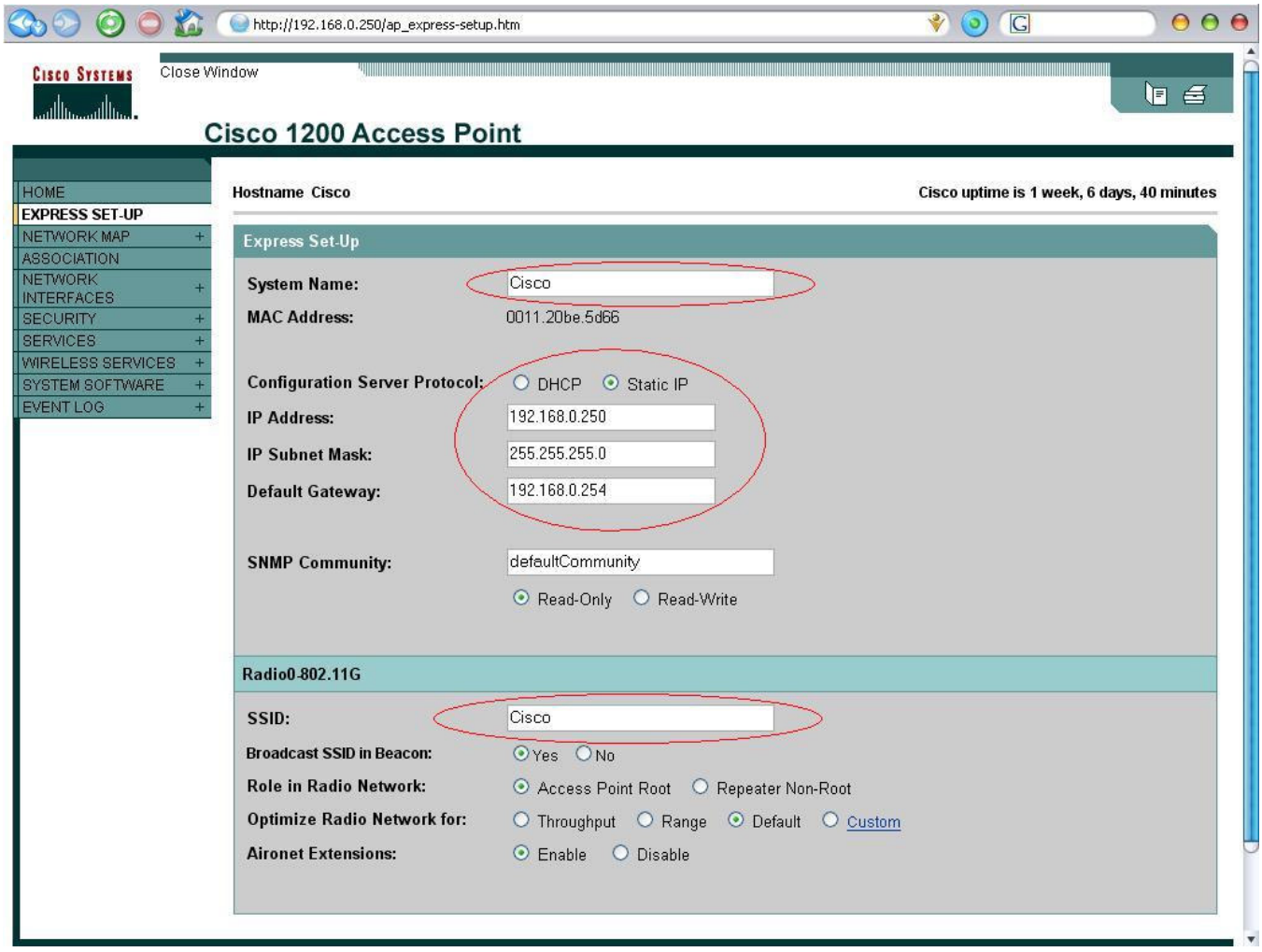
On se connecte au point d'accès via le port de console :

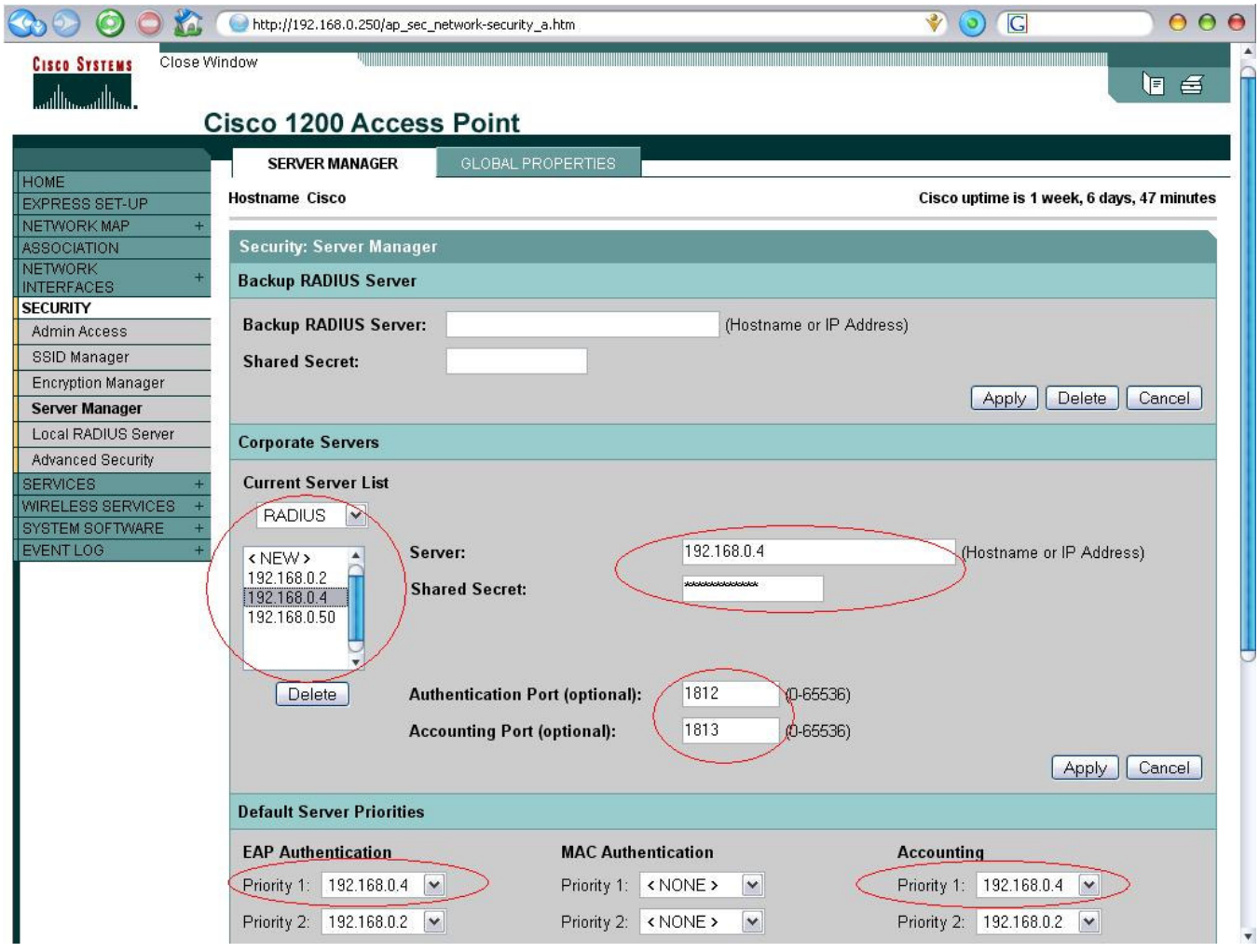
```
Cisco>enable
Password:
Cisco#conf t
Cisco(config)#int f 0
Cisco(config-if)#ip address 192.168.0.10 255.255.255.0
Cisco(config-if)#no sh
Cisco(config-if)#exit
Cisco(config)#int d 0
Cisco(config-if)#ip address 192.168.0.250 255.255.255.0
Cisco(config-if)#no sh
Cisco(config-if)#exit
Cisco(config)#int b 1
Cisco(config-if)#ip address 192.168.0.250 255.255.255.0
Cisco(config-if)#no sh
Cisco(config-if)#exit
```

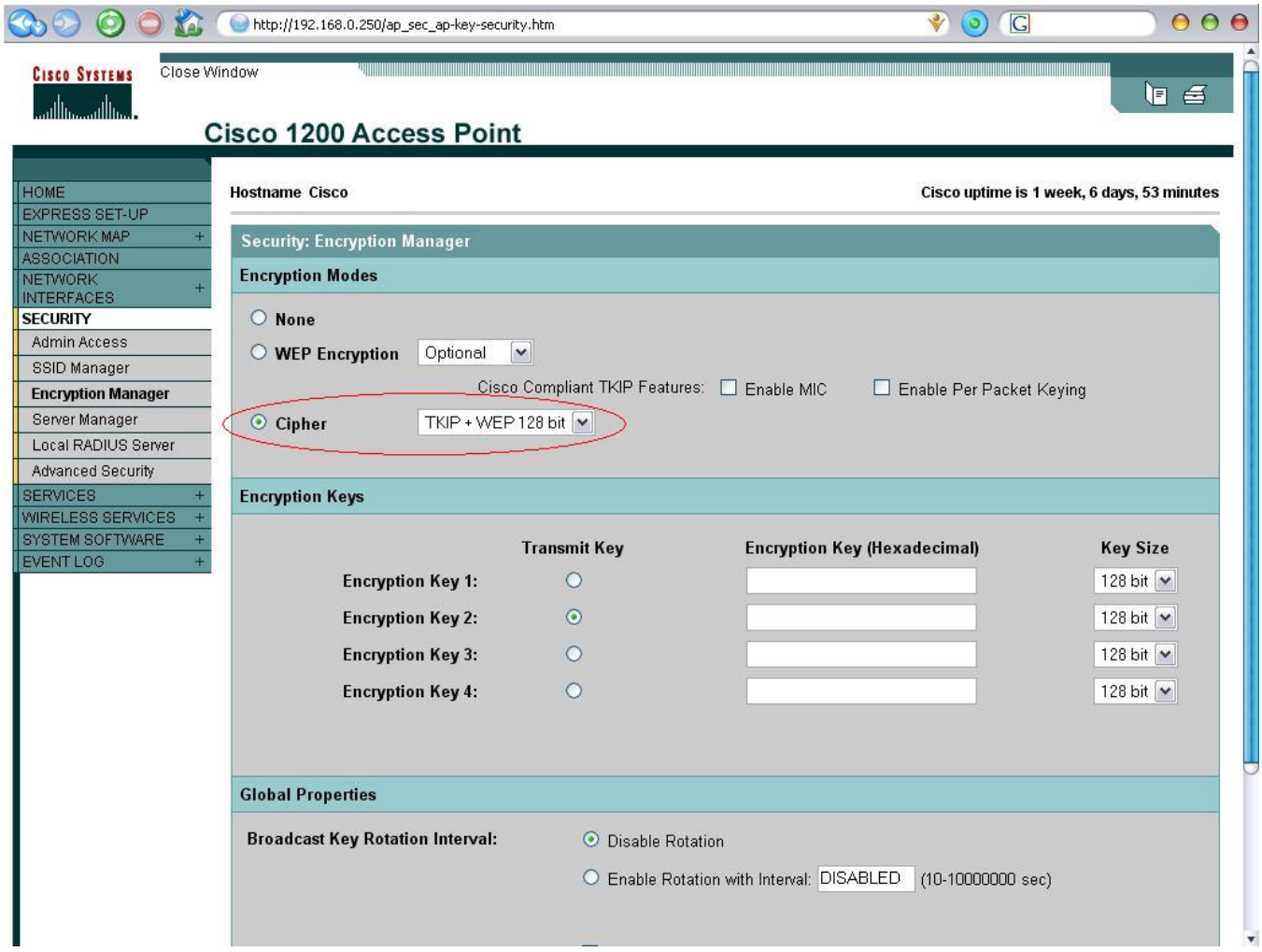
Désormais, on peut, soit continuer à configurer notre point d'accès en console, soit via l'interface web.

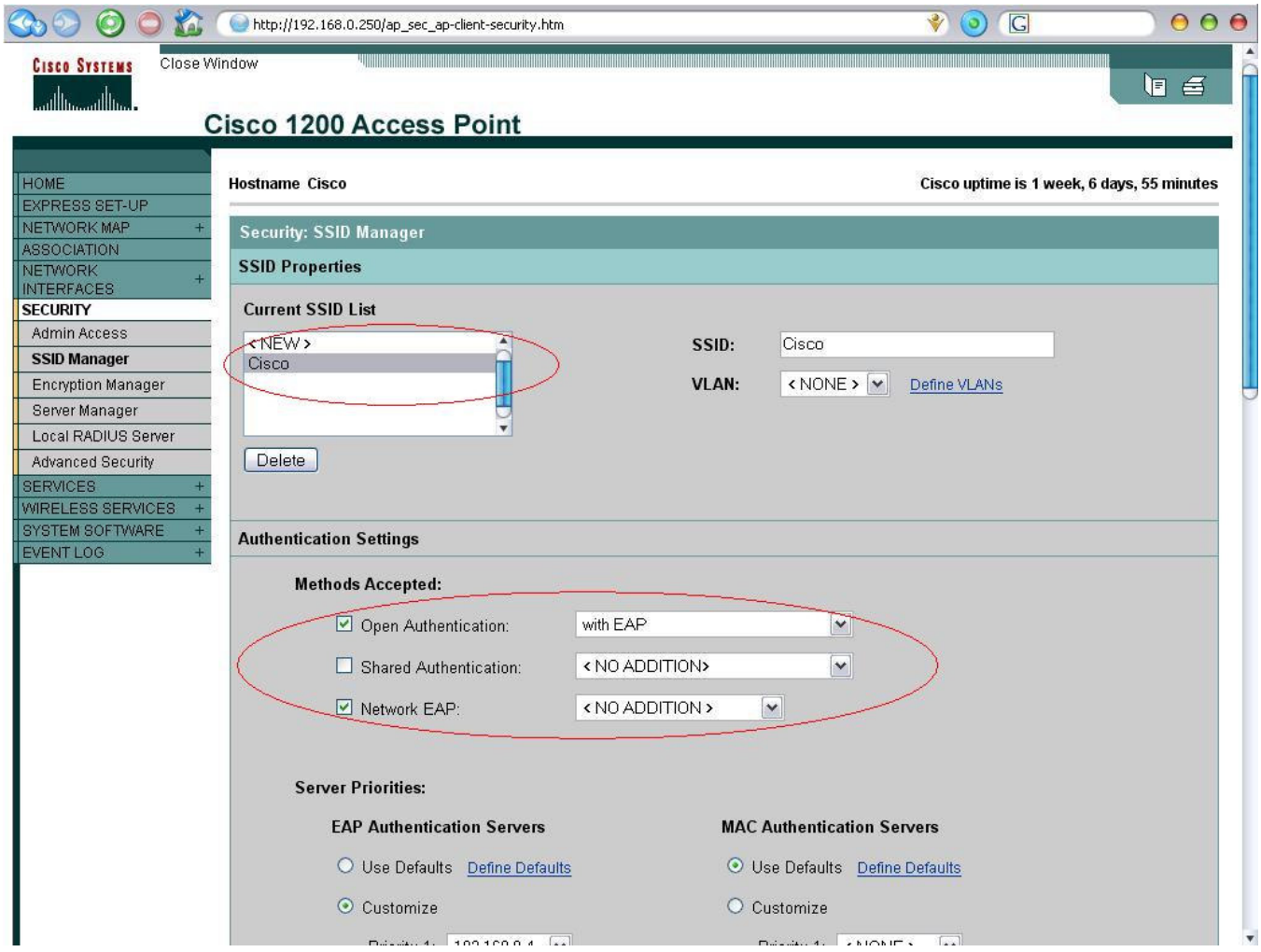
Interface web du point d'accès (les modifications a effectuées sont entourées en rouge) :

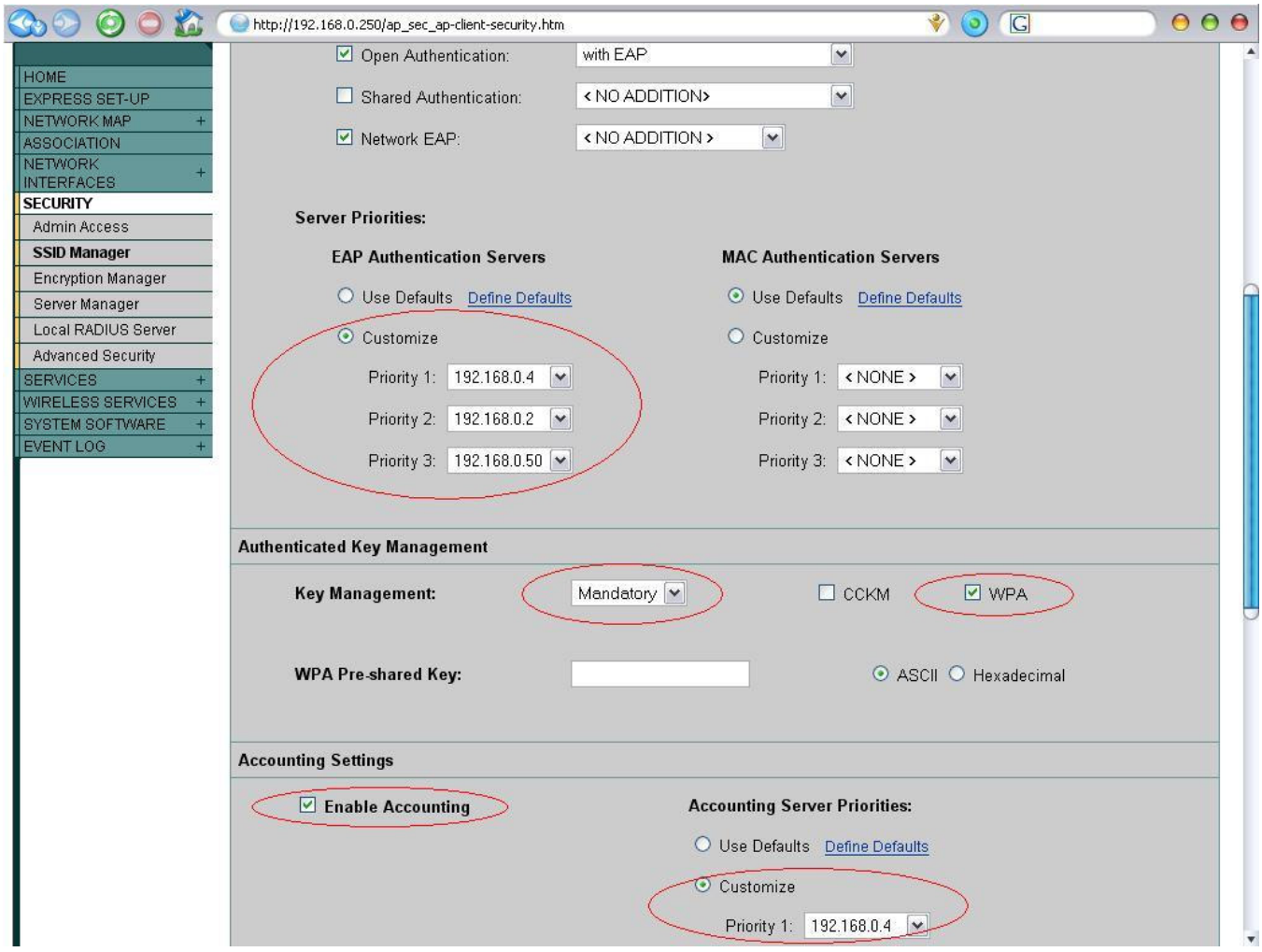


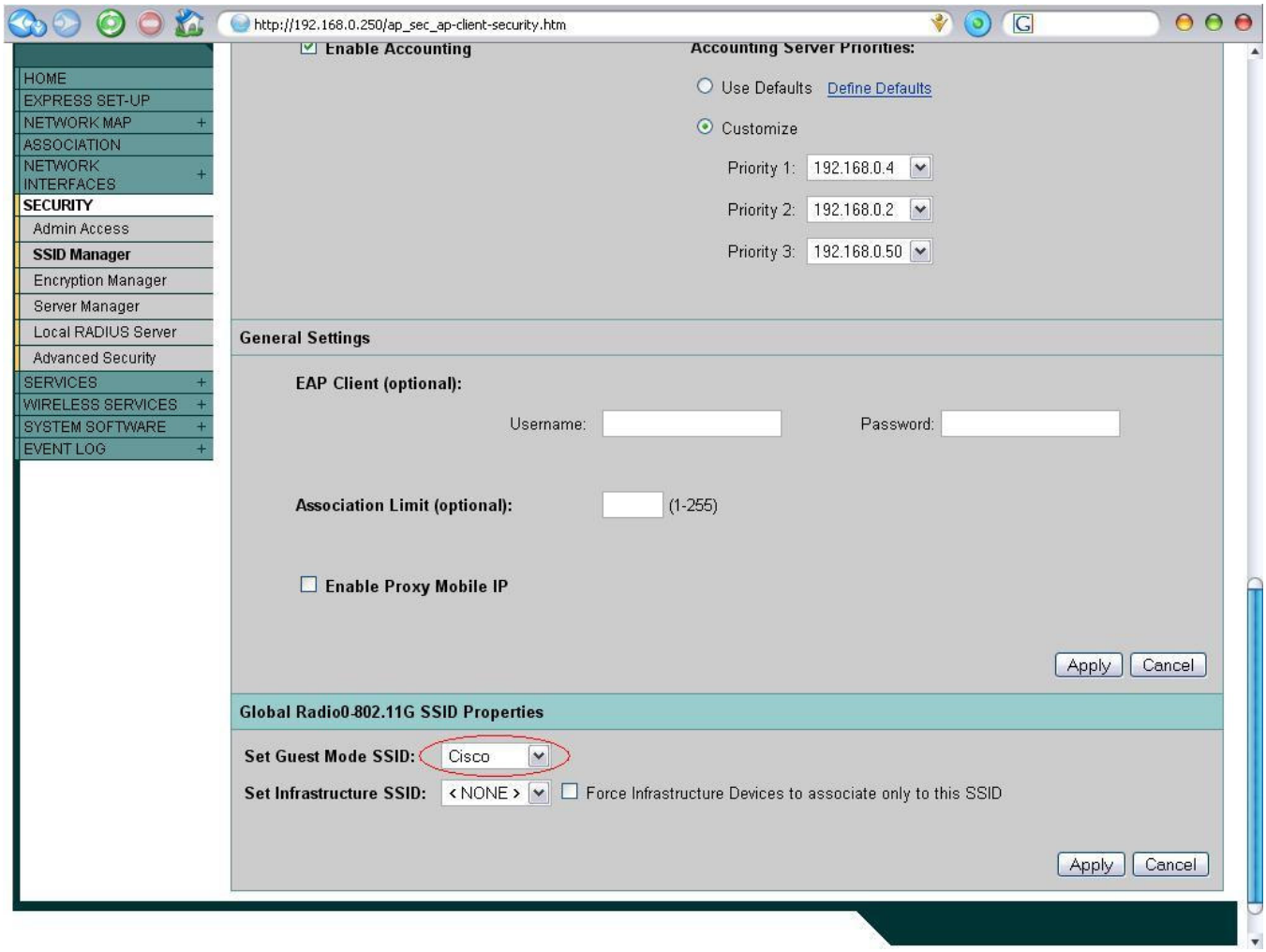

http://192.168.0.250/ap_express-setup.htm
Cisco Systems
Close Window
Cisco 1200 Access Point
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Hostname Cisco
Cisco uptime is 1 week, 6 days, 40 minutes
Express Set-Up
System Name: Cisco
MAC Address: 0011.20be.5d66
Configuration Server Protocol: DHCP Static IP
IP Address: 192.168.0.250
IP Subnet Mask: 255.255.255.0
Default Gateway: 192.168.0.254
SNMP Community: defaultCommunity
Read-Only Read-Write
Radio0-802.11G
SSID: Cisco
Broadcast SSID in Beacon: Yes No
Role in Radio Network: Access Point Root Repeater Non-Root
Optimize Radio Network for: Throughput Range Default Custom
Aironet Extensions: Enable Disable


http://192.168.0.250/ap_sec_network-security_a.htm
Cisco Systems
Close Window
Cisco 1200 Access Point
SERVER MANAGER
GLOBAL PROPERTIES
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
Admin Access
SSID Manager
Encryption Manager
Server Manager
Local RADIUS Server
Advanced Security
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Hostname Cisco
Cisco uptime is 1 week, 6 days, 47 minutes
Security: Server Manager
Backup RADIUS Server
Backup RADIUS Server: (Hostname or IP Address)
Shared Secret:
Apply Delete Cancel
Corporate Servers
Current Server List
RADIUS
< NEW >
192.168.0.2
192.168.0.4
192.168.0.50
Delete
Server: 192.168.0.4 (Hostname or IP Address)
Shared Secret:
Authentication Port (optional): 1812 (0-65536)
Accounting Port (optional): 1813 (0-65536)
Apply Cancel
Default Server Priorities
EAP Authentication
Priority 1: 192.168.0.4
Priority 2: 192.168.0.2
MAC Authentication
Priority 1: < NONE >
Priority 2: < NONE >
Accounting
Priority 1: 192.168.0.4
Priority 2: 192.168.0.2


http://192.168.0.250/ap_sec_ap-key-security.htm
Cisco Systems
Close Window
Cisco 1200 Access Point
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
Admin Access
SSID Manager
Encryption Manager
Server Manager
Local RADIUS Server
Advanced Security
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Hostname Cisco
Cisco uptime is 1 week, 6 days, 53 minutes
Security: Encryption Manager
Encryption Modes
None
WEP Encryption
Optional
Cisco Compliant TKIP Features:
Enable MIC
Enable Per Packet Keying
Cipher
TKIP + WEP 128 bit
Encryption Keys
Transmit Key
Encryption Key (Hexadecimal)
Key Size
Encryption Key 1:
Encryption Key 2:
Encryption Key 3:
Encryption Key 4:
128 bit
Global Properties
Broadcast Key Rotation Interval:
Disable Rotation
Enable Rotation with Interval:
DISABLED
(10-10000000 sec)


http://192.168.0.250/ap_sec_ap-client-security.htm
Cisco Systems
Close Window
Cisco 1200 Access Point
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
Admin Access
SSID Manager
Encryption Manager
Server Manager
Local RADIUS Server
Advanced Security
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Hostname Cisco
Cisco uptime is 1 week, 6 days, 55 minutes
Security: SSID Manager
SSID Properties
Current SSID List
< NEW >
Cisco
SSID:
Cisco
VLAN:
< NONE >
Define VLANs
Delete
Authentication Settings
Methods Accepted:
Open Authentication:
with EAP
Shared Authentication:
< NO ADDITION>
Network EAP:
< NO ADDITION >
Server Priorities:
EAP Authentication Servers
MAC Authentication Servers
Use Defaults
Define Defaults
Customize
Use Defaults
Define Defaults
Customize


http://192.168.0.250/ap_sec_ap-client-security.htm
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
Admin Access
SSID Manager
Encryption Manager
Server Manager
Local RADIUS Server
Advanced Security
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Open Authentication: with EAP
Shared Authentication: < NO ADDITION>
Network EAP: < NO ADDITION >
Server Priorities:
EAP Authentication Servers
Use Defaults Define Defaults
Customize
Priority 1: 192.168.0.4
Priority 2: 192.168.0.2
Priority 3: 192.168.0.50
MAC Authentication Servers
Use Defaults Define Defaults
Customize
Priority 1: < NONE >
Priority 2: < NONE >
Priority 3: < NONE >
Authenticated Key Management
Key Management: Mandatory
CCKM
WPA
WPA Pre-shared Key:
ASCII Hexadecimal
Accounting Settings
Enable Accounting
Accounting Server Priorities:
Use Defaults Define Defaults
Customize
Priority 1: 192.168.0.4

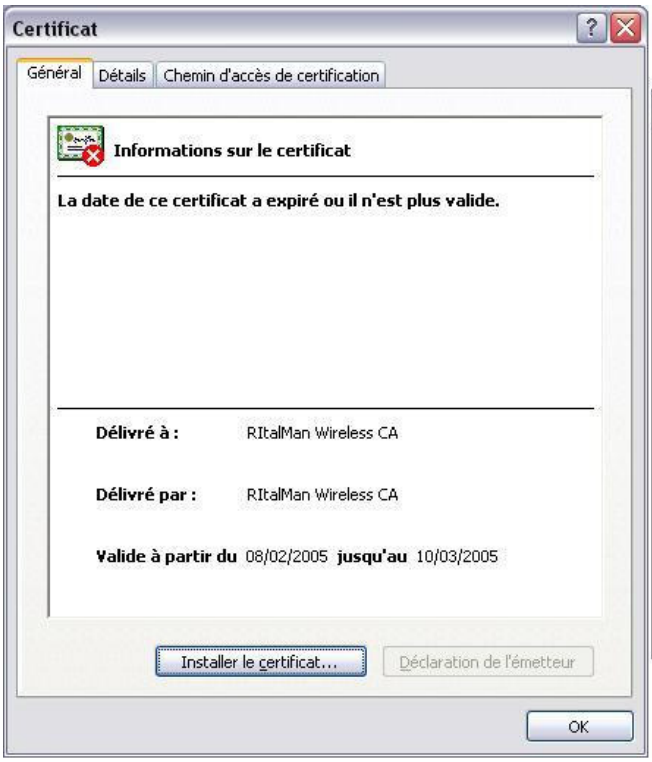

http://192.168.0.250/ap_sec_ap-client-security.htm
HOME
EXPRESS SET-UP
NETWORK MAP
ASSOCIATION
NETWORK INTERFACES
SECURITY
Admin Access
SSID Manager
Encryption Manager
Server Manager
Local RADIUS Server
Advanced Security
SERVICES
WIRELESS SERVICES
SYSTEM SOFTWARE
EVENT LOG
Enable Accounting
Accounting Server Priorities:
Use Defaults Define Defaults
Customize
Priority 1: 192.168.0.4
Priority 2: 192.168.0.2
Priority 3: 192.168.0.50
General Settings
EAP Client (optional):
Username:
Password:
Association Limit (optional):
(1-255)
Enable Proxy Mobile IP
Apply
Cancel
Global Radio0-802.11G SSID Properties
Set Guest Mode SSID: Cisco
Set Infrastructure SSID: < NONE >
Force Infrastructure Devices to associate only to this SSID
Apply
Cancel

C. Côté client

- ✓ Windows XP SP2
  - Installation du certificat

Il faut disposer des certificats créés auparavant (**root.pem, root.der, root.p12**).

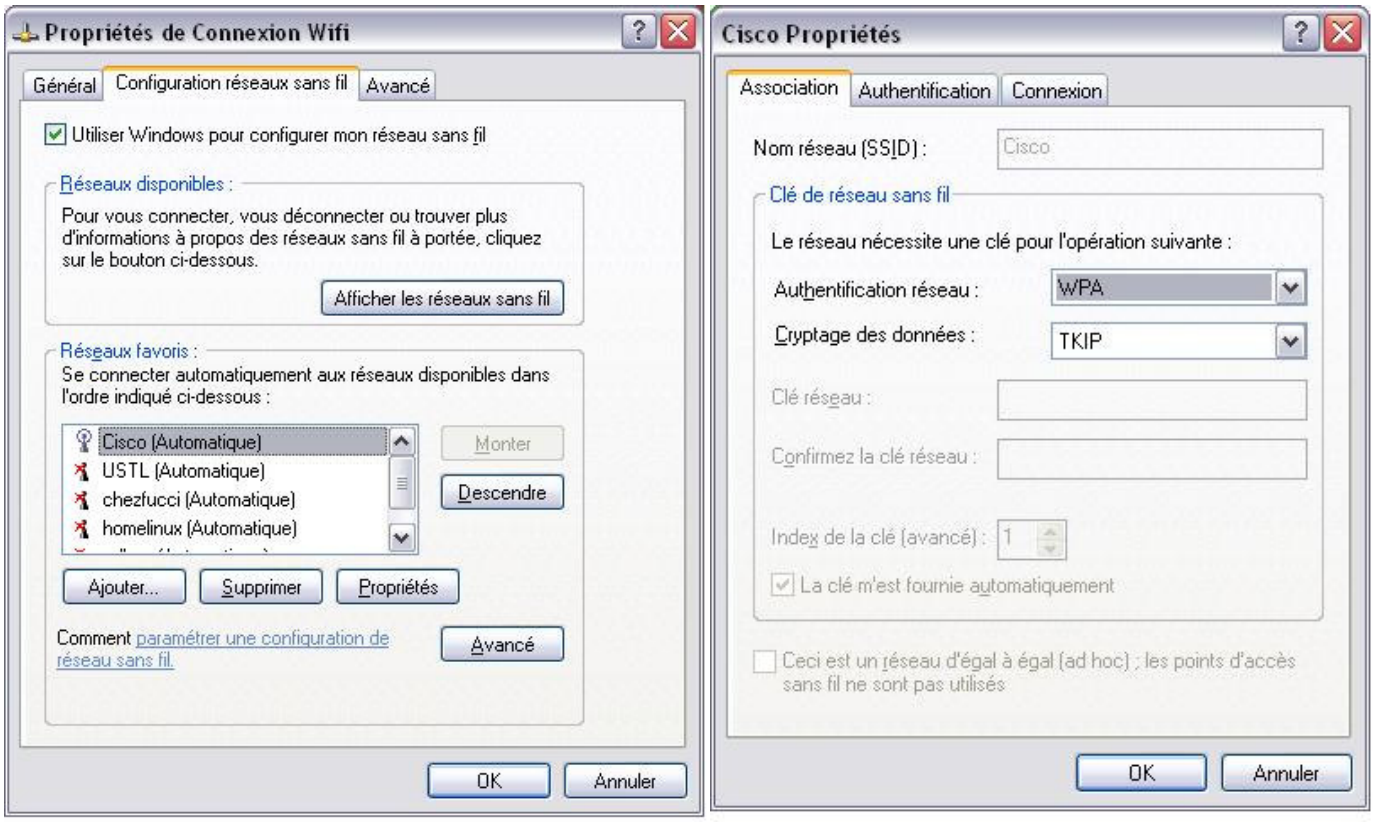
Il faut double cliquer sur le fichier **root.der**, puis sur installer le certificat :

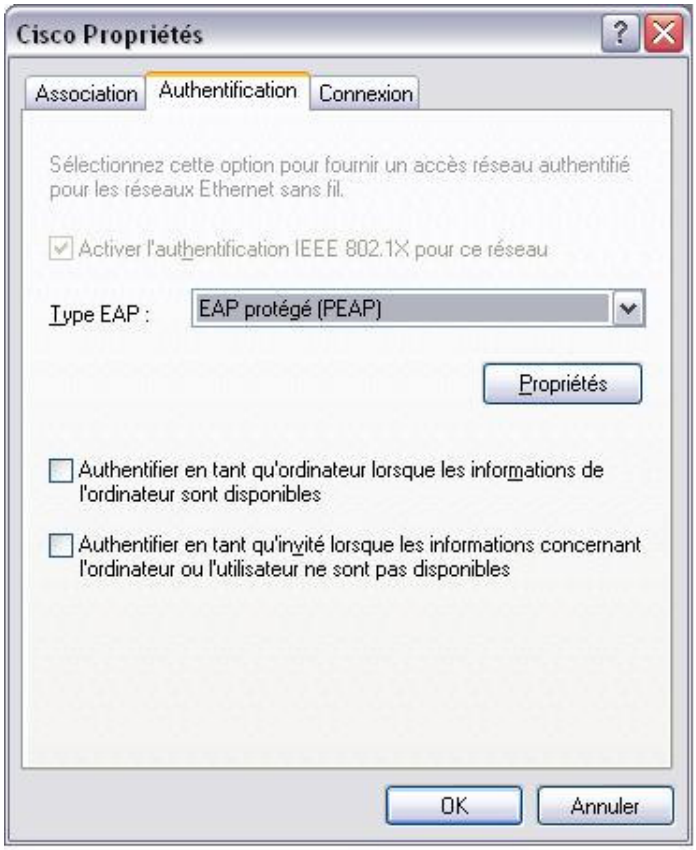


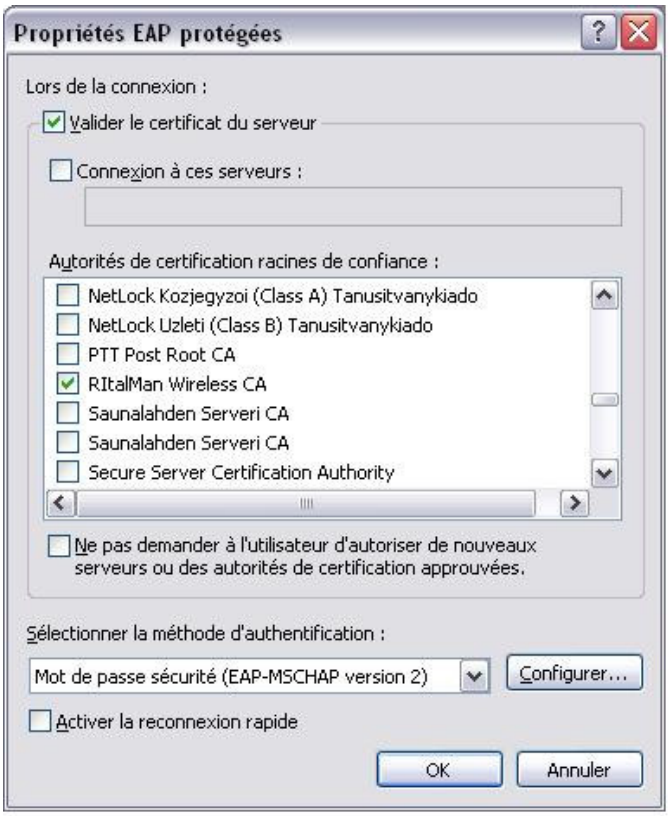
Sélectionnez le magasin **Autorités de certification racines de confiance**.

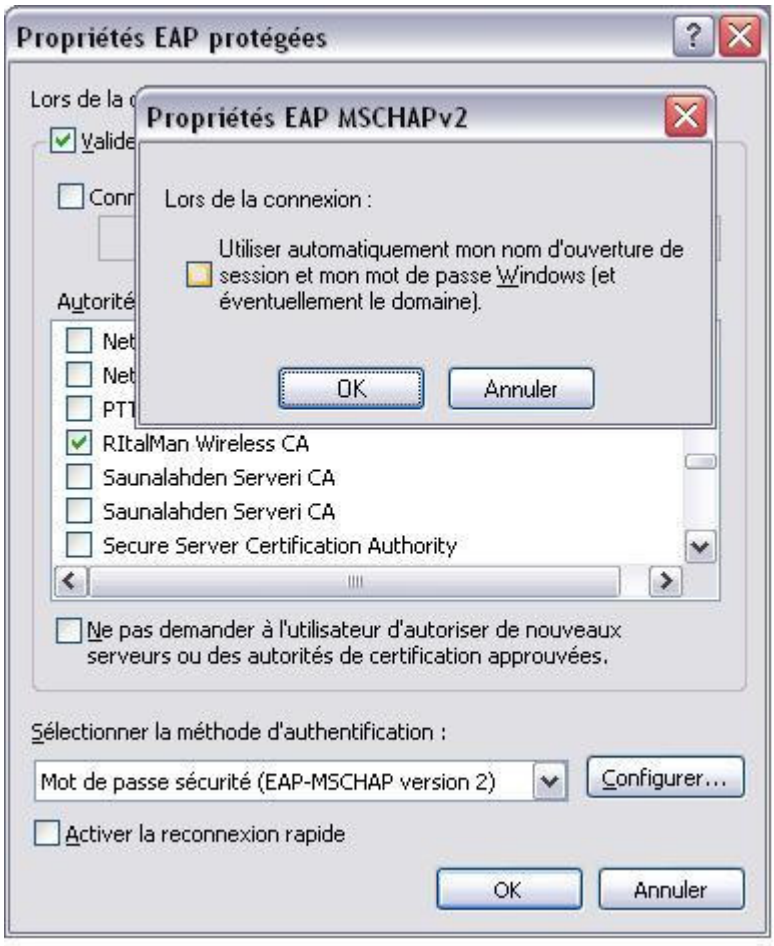
- Configuration de la carte wifi :

Il faut éditer les propriétés de la carte wifi :

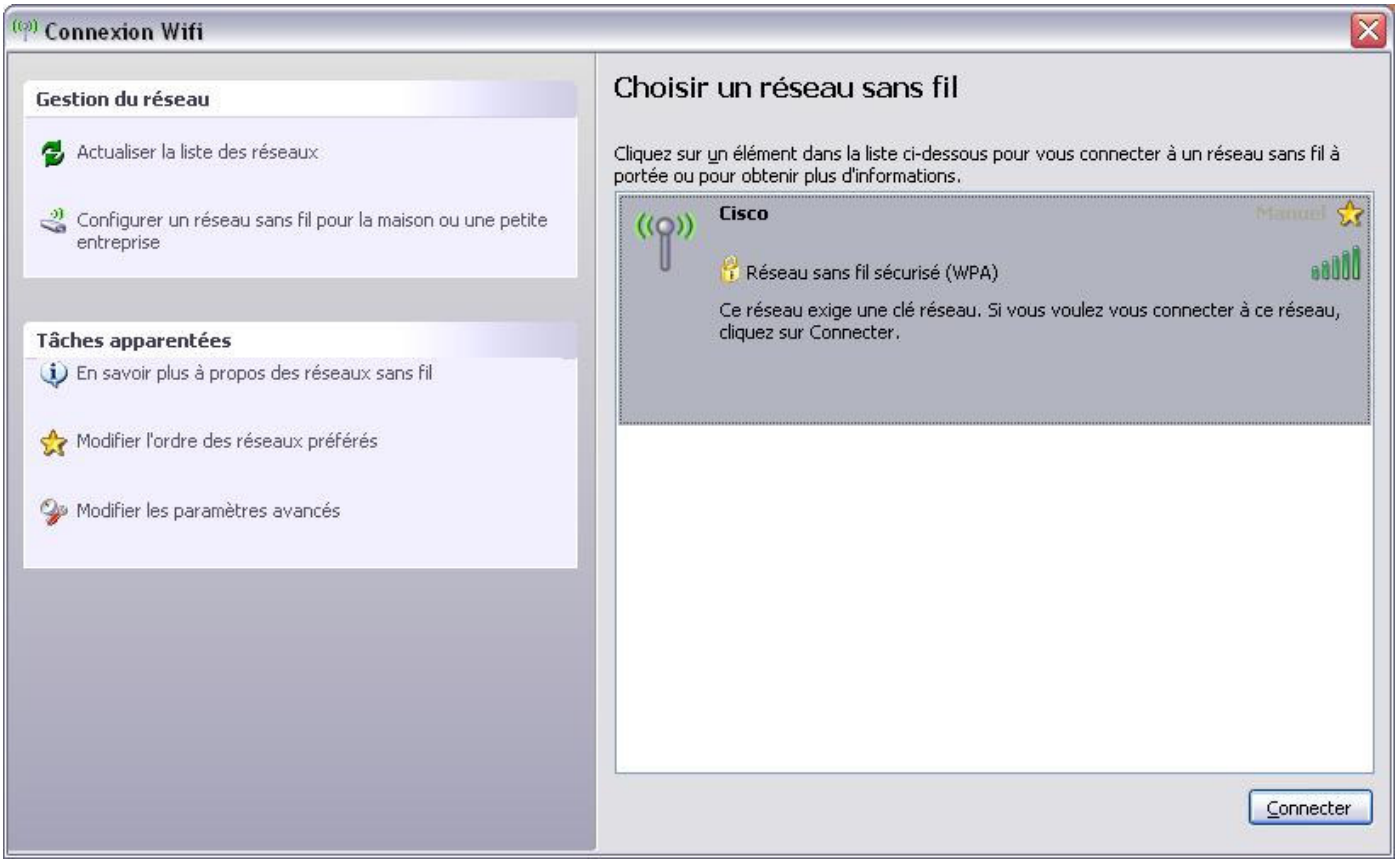



Cisco Propriétés
Association
Authentification
Connexion
Sélectionnez cette option pour fournir un accès réseau authentifié pour les réseaux Ethernet sans fil.
Activer l'authentification IEEE 802.1X pour ce réseau
Type EAP :
EAP protégé (PEAP)
Propriétés
Authentifier en tant qu'ordinateur lorsque les informations de l'ordinateur sont disponibles
Authentifier en tant qu'invité lorsque les informations concernant l'ordinateur ou l'utilisateur ne sont pas disponibles
OK
Annuler

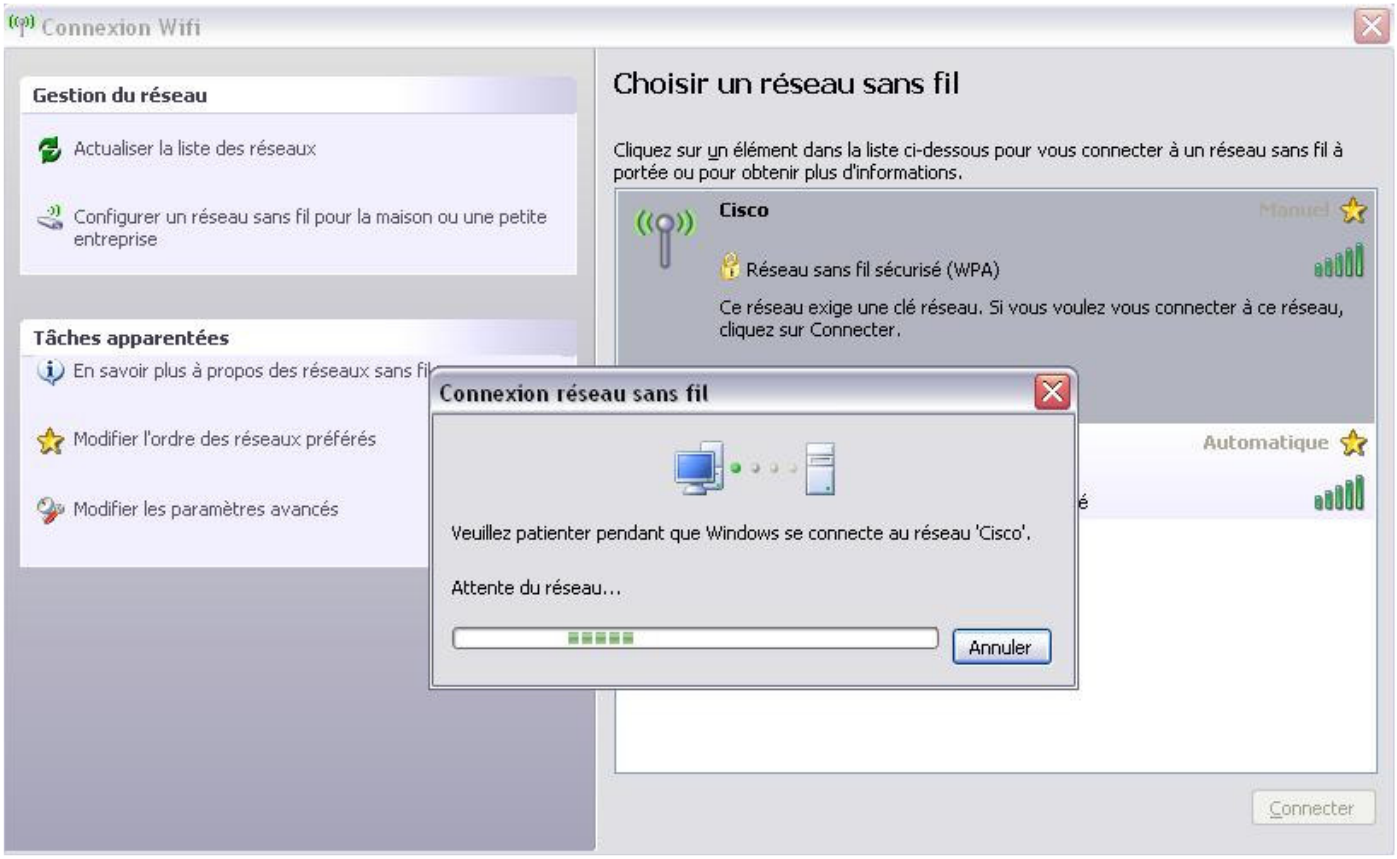

Propriétés EAP protégées
Lors de la connexion :
Valider le certificat du serveur
Connexion à ces serveurs :
Autorités de certification racines de confiance :
NetLock Kozjegyzoi (Class A) Tanusitvanykiado
NetLock Uzleti (Class B) Tanusitvanykiado
PTT Post Root CA
RItalMan Wireless CA
Saunalahden Serveri CA
Saunalahden Serveri CA
Secure Server Certification Authority
Ne pas demander à l'utilisateur d'autoriser de nouveaux serveurs ou des autorités de certification approuvées.
Sélectionner la méthode d'authentification :
Mot de passe sécurité (EAP-MSCHAP version 2)
Configurer...
Activer la reconnexion rapide
OK
Annuler

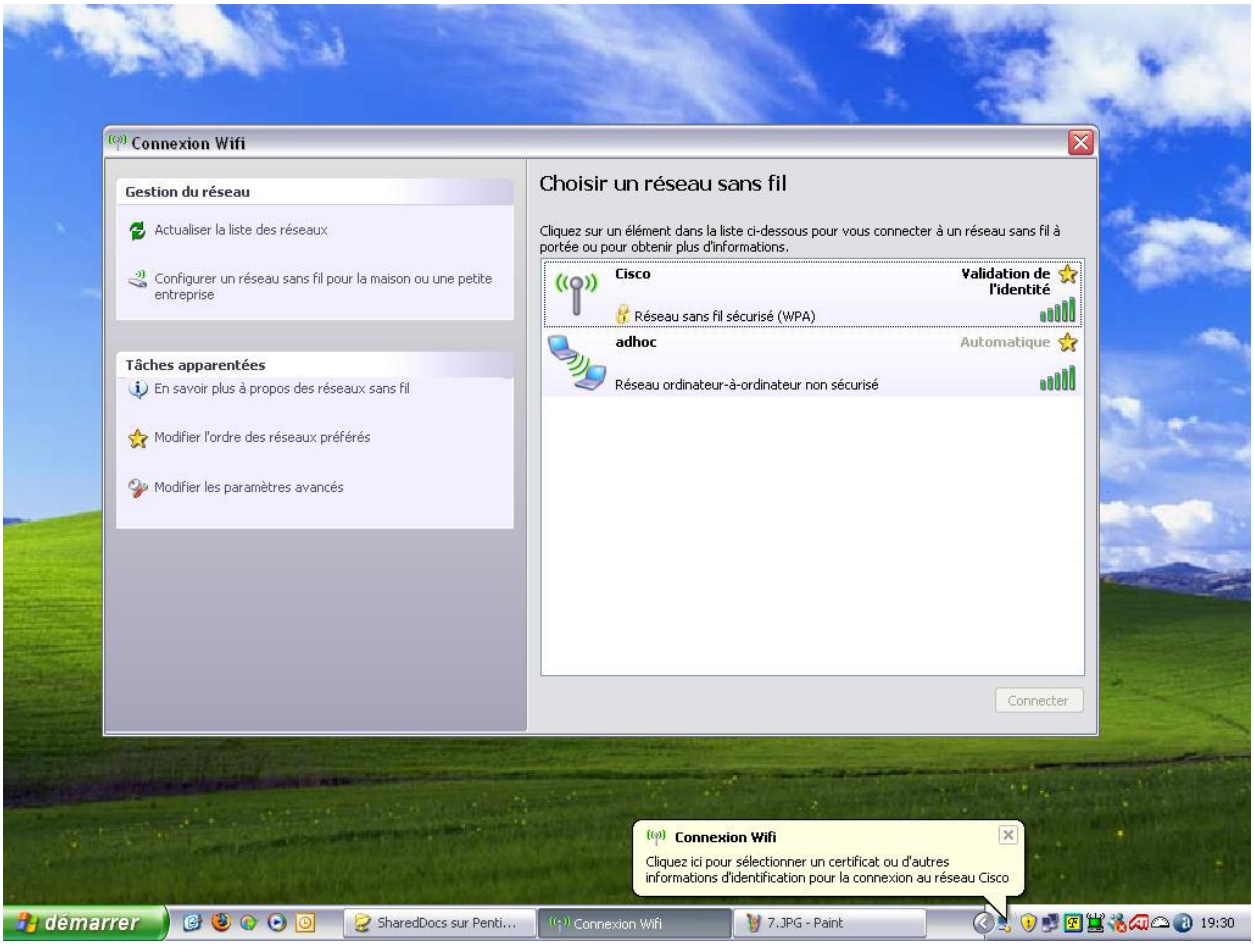

Propriétés EAP protégées
Propriétés EAP MSCHAPv2
Lors de la connexion :
Utiliser automatiquement mon nom d'ouverture de session et mon mot de passe Windows (et éventuellement le domaine).
OK
Annuler
RItalMan Wireless CA
Saunalahden Serveri CA
Saunalahden Serveri CA
Secure Server Certification Authority
Ne pas demander à l'utilisateur d'autoriser de nouveaux serveurs ou des autorités de certification approuvées.
Sélectionner la méthode d'authentification :
Mot de passe sécurité (EAP-MSCHAP version 2)
Configurer...
Activer la reconnexion rapide
OK
Annuler

Nous pouvons désormais nous connecter au réseau sans fil :

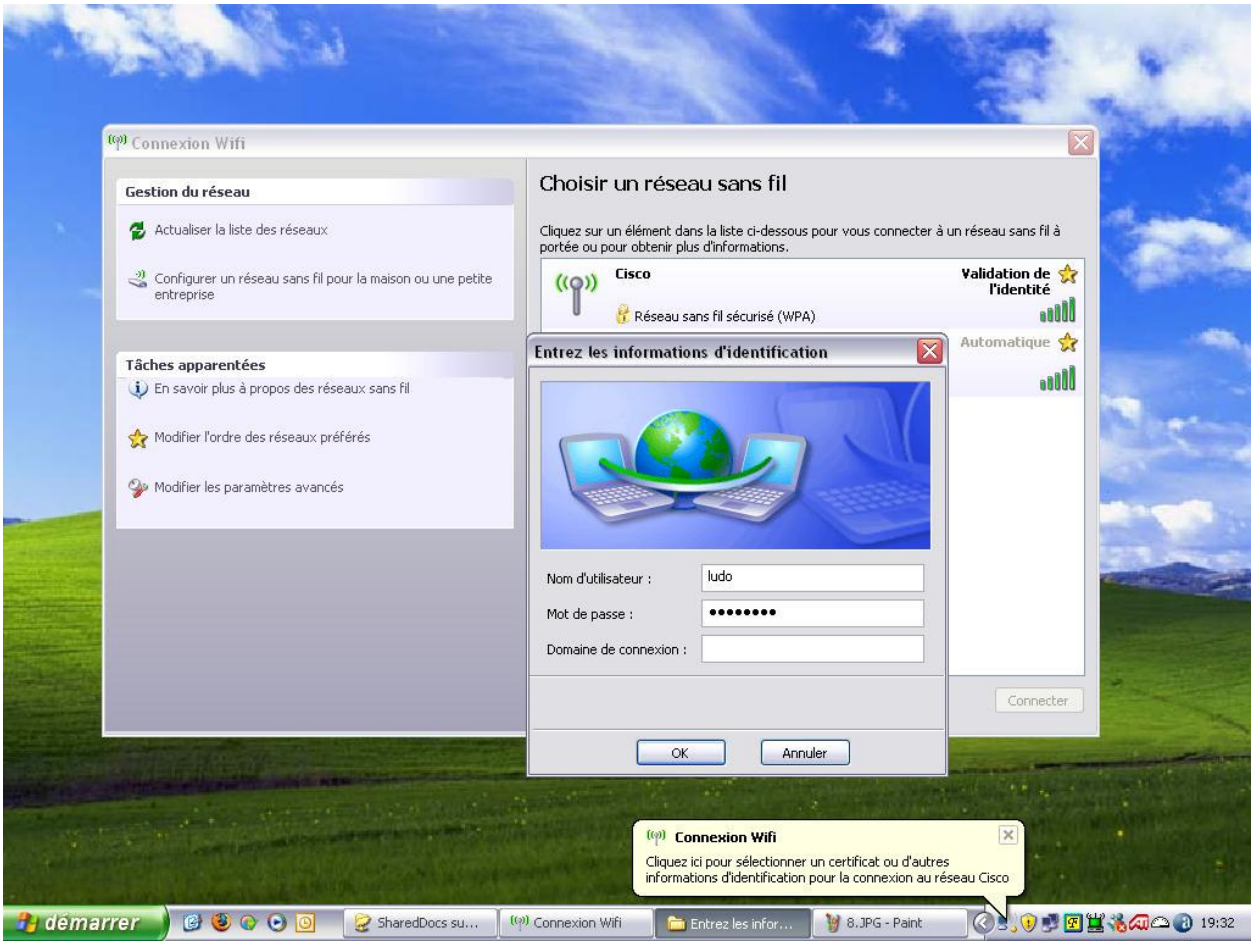


Phase de connexion en cours…

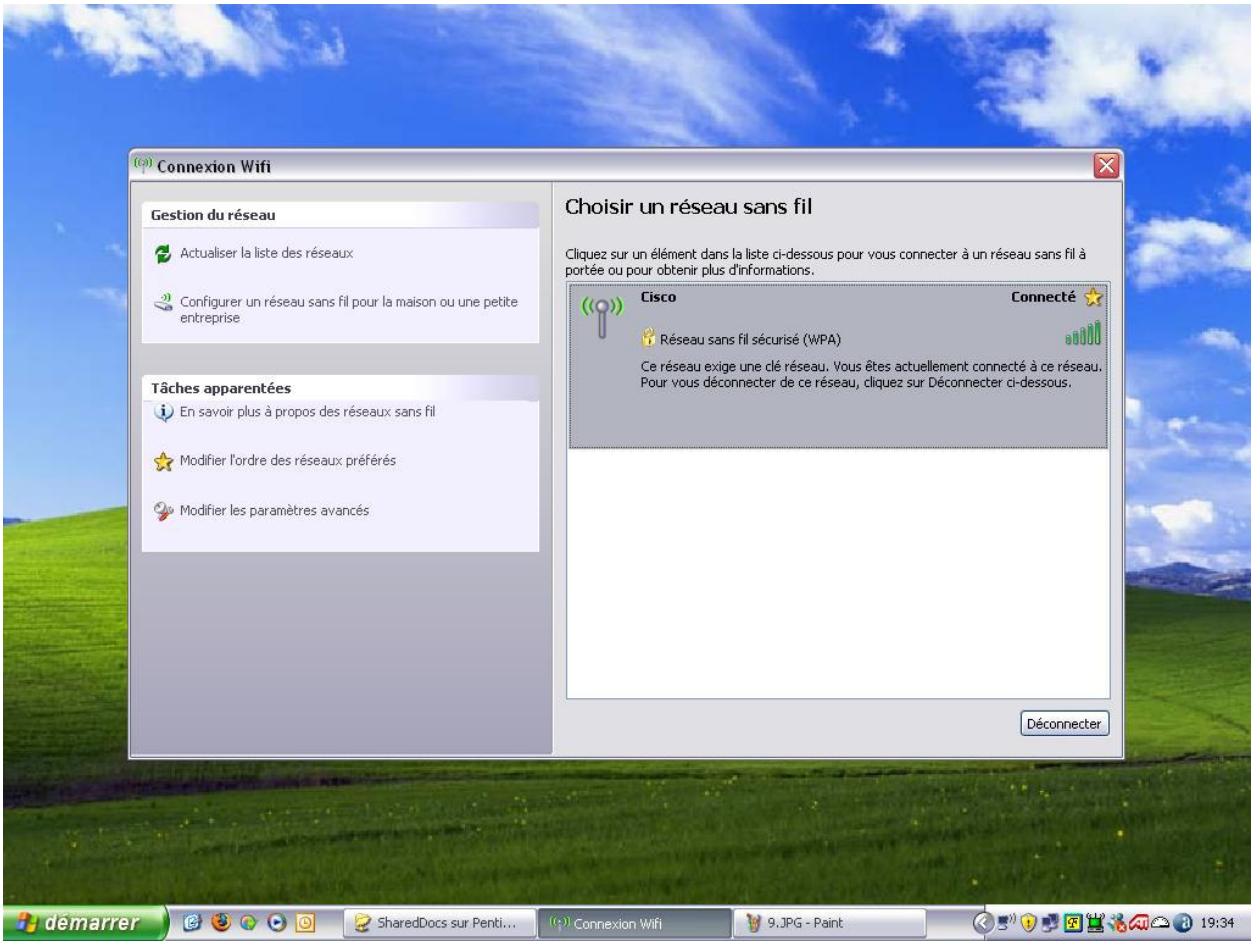




Demande d'authentification :



Nous sommes connectés au réseau wifi.



- ✓ Linux

Il faut disposer des certificats créés auparavant (**root.pem, root.der, root.p12**).

Il nous faut récupérer l'archive du logiciel **wpa_supplicant** et la décompresser :

```
# wget http://hostap.epitest.fi/releases/wpa_supplicant-0.3.8.tar.gz
...
# tar zxvf wpa_supplicant-0.3.8.tar.gz
...
# cd wpa_supplicant-0.3.8.tar.gz
```

Il faut ensuite créer le fichier de configuration **.config** avant la compilation :

```
# Driver interface for Host AP driver
CONFIG_DRIVER_HOSTAP=y

# Driver interface for madwifi driver
CONFIG_DRIVER_MADWIFI=y
# Change include directories to match with the local setup
CFLAGS += -I/usr/src/madwifi-20050228

# Enable IEEE 802.1X Supplicant (automatically included if any EAP method is
# included)
CONFIG_IEEE8021X_EAPOL=y

# EAP-MD5 (automatically included if EAP-TTLS is enabled)
CONFIG_EAP_MD5=y
# EAP-MSCHAPv2 (automatically included if EAP-PEAP is enabled)
CONFIG_EAP_MSCHAPV2=y
# EAP-TLS
CONFIG_EAP_TLS=y
# EAL-PEAP
CONFIG_EAP_PEAP=y
# EAP-TTLS
CONFIG_EAP_TTLS=y
# EAP-GTC
CONFIG_EAP_GTC=y
# EAP-OTP
CONFIG_EAP_OTP=y
# LEAP
CONFIG_EAP_LEAP=y

# PKCS#12 (PFX) support (used to read private key and certificate file from
# a file that usually has extension .p12 or .pfx)
CONFIG_PKCS12=y

# Include control interface for external programs, e.g, wpa_cli
CONFIG_CTRL_IFACE=y
```

Puis compiler le tout :

```
# make && make install
...
```

Une fois **wpa_supplicant** compilé et installé, il faut créer son fichier de configuration **/etc/wpa_supplicant.conf** :

```
network={
      ssid="Cisco"
      key_mgmt=WPA-EAP
      ca_cert="/home/ludo/root.pem"
      identity="ludo"
      password="reseau59"
}
```

On peut maintenant se connecter au réseau sans fil avec la commande suivante :

```
# wpa_supplicant -i ath0 -D madwifi -c /etc/wpa_supplicant.conf &
...
```

- ath0 désigne le nom de notre carte réseau sans fil
- madwifi désigne le driver de notre carte wifi

Il faut maintenant obtenir une adresse IP auprès d'un serveur DHCP :

```
# dhclient ath0
...
```

On peut aussi créer un shell script, **connexion_wifi.sh**, qui automatisera ces commandes :

```
#!/bin/bash

WPA=/usr/local/sbin/wpa_supplicant
CONF=/etc/wpa_supplicant.conf
WLAN=ath0
DRIVER=madwifi
DHCP=/sbin/dhclient

$WPA -i $WLAN -D $DRIVER -c $CONF -Bwq ; sleep 5
$DHCP $WLAN -q

exit
```

```
# ./connexion_wifi.sh
...
```

Conclusion

L'utilisation d'un serveur RADIUS est une excellente solution afin de procéder à une authentification forte des clients.

De plus, RADIUS ne permet pas que l'authentification des clients wifi, il peut également être utilisé dans la phase d'authentification des clients filaires…
Dans ce cas, c'est le switch, par exemple, qui fait office de **NAS** et qui interroge le serveur RADIUS.