Implementación de WAP 1 con Software Libre

Jorge A. Zárate Pérez
Instituto Tecnológico de Oaxaca

jorge.zarate@itoaxaca.edu.mx

INSTALACION DEL RADIUS

```
 ./configure
make
make install
```

Scripts para generar los certificados

CA.clt

```
-------------------------------------------------------------------------
#!/bin/bash
SSL=/usr/local/ssl
export PATH=${SSL}/bin/:${SSL}/ssl/misc:${PATH}
export LD_LIBRARY_PATH=${SSL}/lib
echo
"**********************************************************************"
echo "Creating client private key and certificate"
echo "When prompted enter the client name in the Common Name field. This
is the same"
echo " used as the Username in FreeRADIUS"
echo
"**********************************************************************"
echo
# Request a new PKCS#10 certificate.
# First, newreq.pem will be overwritten with the new certificate request
openssl req -new -keyout newreq.pem -out newreq.pem -passin
pass:whatever -passout pass:whatever
# Sign the certificate request. The policy is defined in the openssl.cnf
file.
# The request generated in the previous step is specified with the -in-
files option and
# the output is in newcert.pem
# The -extensions option is necessary to add the OID for the extended
key for client authentication
openssl ca -policy policy_anything -out newcert.pem -passin
pass:whatever -key whatever -extensions xpclient_ext -extfile xpexten-
sions -infiles newreq.pem
# Create a PKCS#12 file from the new certificate and its private key
found in newreq.pem
# and place in file specified on the command line
openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -out $1.p12
-clcerts -passin pass:whatever -passout pass:whatever
# parse the PKCS#12 file just created and produce a PEM format certifi-
cate and key in certclt.pem
openssl pkcs12 -in $1.p12 -out $1.pem -passin pass:whatever -passout
pass:whatever
# Convert certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in $1.pem -out $1.der
# clean up
rm -rf newcert newreq.pem
```

CA.root

```
-------------------------------------------------------------------------
#!/bin/bash
```

```
SSL=/usr/local/ssl
export PATH=${SSL}/bin/:${SSL}/ssl/misc:${PATH}
export LD_LIBRARY_PATH=${SSL}/lib
# needed if you need to start from scratch otherwise the CA.pl -newca
command doesn't copy the new
# private key into the CA directories
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
# Generate a new self-signed certificate.
# After invocation, newreq.pem will contain a private key and certifi-
cate
# newreq.pem will be used in the next step
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
"**********************************************************************"
echo
# Create a PKCS#12 file, using the previously created CA certificate/key
# The certificate in demoCA/cacert.pem is the same as in newreq.pem. In-
stead of
# using "-in demoCA/cacert.pem" we could have used "-in newreq.pem" and
then omitted
# the "-inkey newreq.pem" because newreq.pem contains both the private
key and certificate
openssl pkcs12 -export -in demoCA/cacert.pem -inkey newreq.pem -out
root.p12 -cacerts -passin pass:whatever -passout pass:whatever
# parse the PKCS#12 file just created and produce a PEM format certifi-
cate and key in root.pem
openssl pkcs12 -in root.p12 -out root.pem -passin pass:whatever -passout
pass:whatever
# Convert root certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in root.pem -out root.der
#Clean Up
rm -rf newreq.pem
```

CA.srv

```
---------------------------------------------------------------------
#!/bin/bash
SSL=/usr/local/ssl
```

```
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
"**********************************************************************"
echo
# Request a new PKCS#10 certificate.
# First, newreq.pem will be overwritten with the new certificate request
openssl req -new -keyout newreq.pem -out newreq.pem -passin
pass:whatever -passout pass:whatever
# Sign the certificate request. The policy is defined in the openssl.cnf
file.
# The request generated in the previous step is specified with the -in-
files option and
# the output is in newcert.pem
# The -extensions option is necessary to add the OID for the extended
key for server authentication
openssl ca -policy policy_anything -out newcert.pem -passin
pass:whatever -key whatever -extensions xpserver_ext -extfile xpexten-
sions -infiles newreq.pem
# Create a PKCS#12 file from the new certificate and its private key
found in newreq.pem
# and place in file specified on the command line
openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -out $1.p12
-clcerts -passin pass:whatever -passout pass:whatever
# parse the PKCS#12 file just created and produce a PEM format certifi-
cate and key in certsrv.pem
openssl pkcs12 -in $1.p12 -out $1.pem -passin pass:whatever -passout
pass:whatever
# Convert certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in $1.pem -out $1.der
# Clean Up
rm -rf newert.pem newreq.pem
```

xpextensions

```
-------------------------------------------------------------------------
extendedKeyUsage = 1.3.6.1.5.5.7.3.1
```

Genracion de Certificados

```
bash-2.03# ./CA.root
**********************************************************************
Creating self-signed private key and certificate
When prompted override the default value for the Common Name field
**********************************************************************

Generating a 1024 bit RSA private key
.........................................++++++
.....................++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
```

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Oaxaca
Locality Name (eg, city) []:Oaxaca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Instituto
Tecnologico de Oaxaca
Organizational Unit Name (eg, section) []:NOCITOX
Common Name (eg, YOUR name) []:ITO
Email Address []:netadmin@itoaxaca.edu.mx
**********************************************************************
Creating a new CA hierarchy (used later by the ca command) with the cer-
tificate
and private key created in the last step
**********************************************************************

**********************************************************************
Creating ROOT CA
**********************************************************************


MAC verified OK
bash-2.03# ls -al
total 32
drwxr-xr-x   3 root     other        512 Apr 13 08:55 .
drwxr-xr-x   4 root     other       1024 Apr 13 08:49 ..
-rwxr-xr-x   1 root     other       1765 Apr 13 08:52 CA.clt
-rwxr-xr-x   1 root     other       2204 Apr 13 08:52 CA.root
-rwxr-xr-x   1 root     other       1672 Apr 13 08:52 CA.svr
drwxr-xr-x   6 root     other        512 Apr 13 08:55 demoCA
-rw-r--r--   1 root     other        981 Apr 13 08:55 root.der
-rw-r--r--   1 root     other       1997 Apr 13 08:55 root.p12
-rw-r--r--   1 root     other       2807 Apr 13 08:55 root.pem

bash-2.03# ./CA.svr rs.itoaxaca.edu.mx
**********************************************************************
Creating server private key and certificate
When prompted enter the server name in the Common Name field.
**********************************************************************

Generating a 1024 bit RSA private key
.......++++++
.............++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Oaxaca
Locality Name (eg, city) []:Oaxaca
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Instituto
Tecnologico de Oaxaca
Organizational Unit Name (eg, section) []:NOCITOX
Common Name (eg, YOUR name) []:ITO
Email Address []:netadmin@itoaxaca.edu.mx

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:t3mp0r4l
An optional company name []:
Using configuration from /usr/local/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            f2:b0:d7:b0:ee:0a:34:50
        Validity
            Not Before: Apr 13 13:57:38 2005 GMT
            Not After : Jan  7 13:57:38 2009 GMT
        Subject:
            countryName               = MX
            stateOrProvinceName       = Oaxaca
            localityName              = Oaxaca
            organizationName          = Instituto Tecnologico de Oaxaca
            organizationalUnitName    = NOCITOX
            commonName                = ITO
            emailAddress              = netadmin@itoaxaca.edu.mx
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Jan  7 13:57:38 2009 GMT (1365
days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
MAC verified OK

bash-2.03# ls -al
total 52
drwxr-xr-x   3 root     other        512 Apr 13 08:57 .
drwxr-xr-x   4 root     other       1024 Apr 13 08:49 ..
-rwxr-xr-x   1 root     other       1765 Apr 13 08:52 CA.clt
-rwxr-xr-x   1 root     other       2204 Apr 13 08:52 CA.root
-rwxr-xr-x   1 root     other       1672 Apr 13 08:52 CA.svr
drwxr-xr-x   6 root     other        512 Apr 13 08:57 demoCA
-rw-r--r--   1 root     other       2980 Apr 13 08:57 newcert.pem
-rw-r--r--   1 root     other        981 Apr 13 08:55 root.der
-rw-r--r--   1 root     other       1997 Apr 13 08:55 root.p12
-rw-r--r--   1 root     other       2807 Apr 13 08:55 root.pem
-rw-r--r--   1 root     other        735 Apr 13 08:57
rs.itoaxaca.edu.mx.der
-rw-r--r--   1 root     other       1749 Apr 13 08:57
rs.itoaxaca.edu.mx.p12
-rw-r--r--   1 root     other       2474 Apr 13 08:57
rs.itoaxaca.edu.mx.pem
-rw-r--r--   1 root     other        107 Apr 13 08:57 xpextensions
```

```
bash-2.03# ./CA.clt users
**********************************************************************
Creating client private key and certificate
When prompted enter the client name in the Common Name field. This is
the same
 used as the Username in FreeRADIUS
**********************************************************************

Generating a 1024 bit RSA private key
.....................................................................++++++
.........................++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Oaxaca
Locality Name (eg, city) []:Oaxaca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Instituto
Tecnologico de Oaxaca
Organizational Unit Name (eg, section) []:NOCITOX
Common Name (eg, YOUR name) []:ITO
Email Address []:netadmin@itoaxaca.edu.mx

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/local/ssl/openssl.cnf
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            f2:b0:d7:b0:ee:0a:34:51
        Validity
            Not Before: Apr 13 14:00:25 2005 GMT
            Not After : Jan  7 14:00:25 2009 GMT
        Subject:
            countryName               = MX
            stateOrProvinceName       = Oaxaca
            localityName              = Oaxaca
            organizationName          = Instituto Tecnologico de Oaxaca
            organizationalUnitName    = NOCITOX
            commonName                = ITO
            emailAddress              = netadmin@itoaxaca.edu.mx
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Jan  7 14:00:25 2009 GMT (1365
days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
unable to load 'random state'
This means that the random number generator has not been seeded
with much random data.
Consider setting the RANDFILE environment variable to point at a file
that
'random' data can be kept in (the file will be overwritten).
MAC verified OK
bash-2.03# ls -al
total 64
drwxr-xr-x   3 root     other        512 Apr 13 09:00 .
drwxr-xr-x   4 root     other       1024 Apr 13 08:49 ..
-rwxr-xr-x   1 root     other       1765 Apr 13 08:52 CA.clt
-rwxr-xr-x   1 root     other       2204 Apr 13 08:52 CA.root
-rwxr-xr-x   1 root     other       1672 Apr 13 08:52 CA.svr
drwxr-xr-x   6 root     other        512 Apr 13 09:00 demoCA
-rw-r--r--   1 root     other       2980 Apr 13 09:00 newcert.pem
-rw-r--r--   1 root     other        981 Apr 13 08:55 root.der
-rw-r--r--   1 root     other       1997 Apr 13 08:55 root.p12
-rw-r--r--   1 root     other       2807 Apr 13 08:55 root.pem
-rw-r--r--   1 root     other        735 Apr 13 08:57
rs.itoaxaca.edu.mx.der
-rw-r--r--   1 root     other       1749 Apr 13 08:57
rs.itoaxaca.edu.mx.p12
-rw-r--r--   1 root     other       2474 Apr 13 08:57
rs.itoaxaca.edu.mx.pem
-rw-r--r--   1 root     other        735 Apr 13 09:00 users.der
-rw-r--r--   1 root     other       1749 Apr 13 09:00 users.p12
-rw-r--r--   1 root     other       2474 Apr 13 09:00 users.pem
-rw-r--r--   1 root     other        107 Apr 13 08:57 xpextensions
```

Configuracion del Servidor Radius

clients.conf

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

users

```
 "mobile" Auth-Type := EAP, User-Password =="test"
```

eap.conf

```
eap {
         default_eap_type = peap
```

```
    ......
     tls {
             private_key_password = whatever
                   private_key_file =
${raddbdir}/certs/rs.itoaxaca.edu.mx.pem
                  certificate_file =
${raddbdir}/certs/rs.itoaxaca.edu.mx.pem
                  CA_file = ${raddbdir}/certs/demoCA/cacert.pem
                  dh_file = ${raddbdir}/certs/DH
                  random_file = ${raddbdir}/certs/random
                  fragment_size = 1024
                  include_length = yes
      }
     ......
            peap {
                        default_eap_type = mschapv2
            }
       .....
}
```

radiusd.conf

```
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 {
 preprocess
 auth_log
 eap
 files
 }
 ...

 authenticate {
 unix
 eap
 }

 detail auth_log {
                 detailfile = ${radacctdir}/%{Client-IP-Address}/auth-
detail-%Y%m%d
                 detailperm = 0600
         }
```

Correr el Servidor Radius

```
radiusd -X -A &
```

Instalando el servidor NTP

```
wget http://www.eecis.udel.edu/~ntp/ntp_spool/ntp4/ntp-4.2.0.tar.gz
gunzip -c ntp-4.2.0.tar.gz | tar xvf -
cd ntp-4.2.0
make
make install
```

Configurar el ntod.conf

ntpd.conf

```
---------------------------------
driftfile /etc/ntp.drift
server 17.254.0.26 prefer

restrict 192.168.0.0 mask 255.255.255.0 nomodify nopeer
restrict 17.254.0.26 noquery nomodify notrap nopeer
restrict 127.0.0.1 nomodify
---------------------------------
```

Correr el servidor ntpd

```
ntpd -c /etc/ntp.conf
```

Instalando el servidor Syslog-NG

```
wget
http://www.balabit.com/downloads/syslog-ng/1.5/src/syslog-ng-1.5.26.tar.
gz
 gunzip -c syslog-ng-1.5.26.tar.gz | tar xvf -
cd  syslog-ng-1.5.26
./configure
make
make install
```

Conf del Syslog

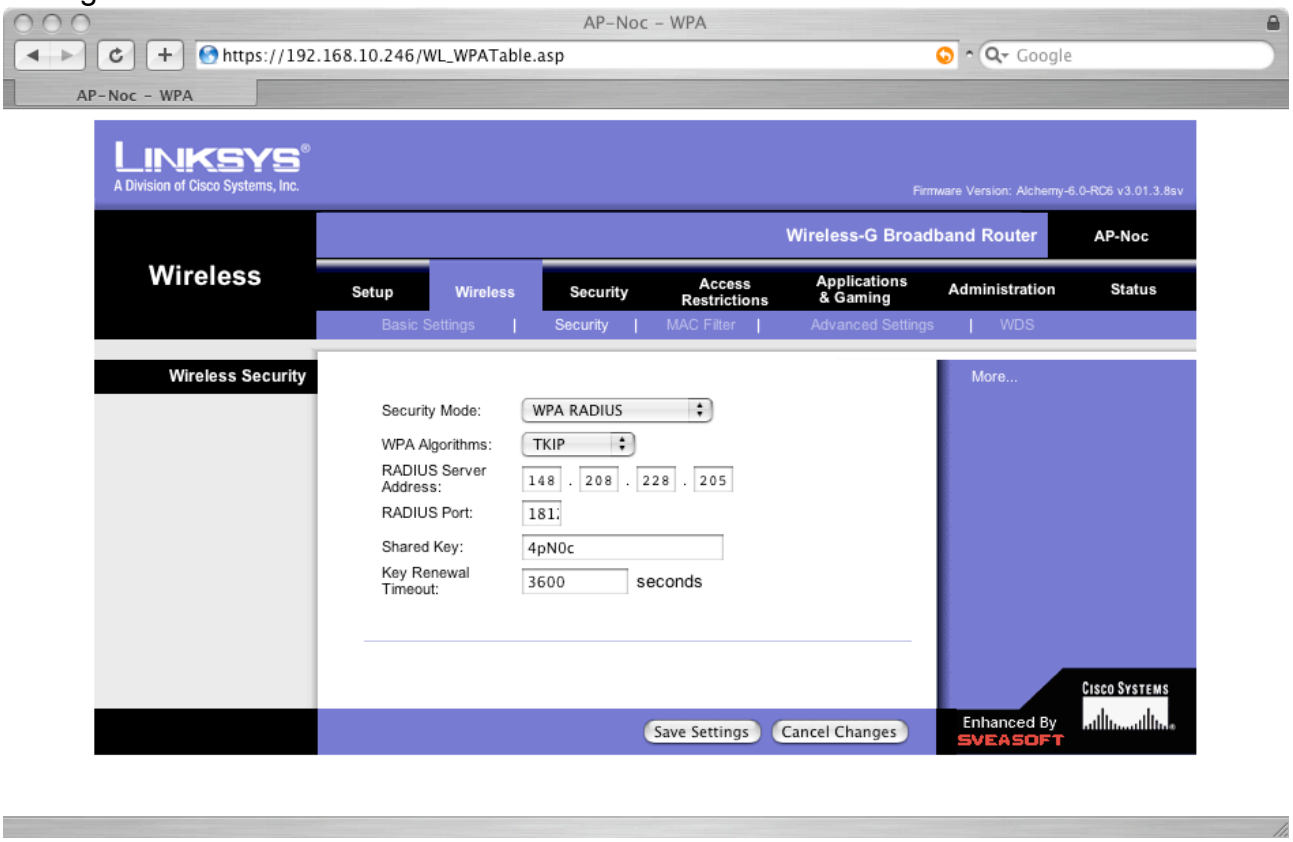
```
cat /etc/syslog-ng/syslog-ng.conf
options {
     use_fqdn(no);
     sync(0);
     };

source s_netAP { udp(); };

destination d_messagesAP { file("/var/log/AP-Noc.log" owner("root")
group("adm") perm(0640)); };

filter f_AP1 { host(192.168.10.246); };

log { source(s_netAP); filter(f_AP1); destination(d_messagesAP); };
```

correr el syslog-ng

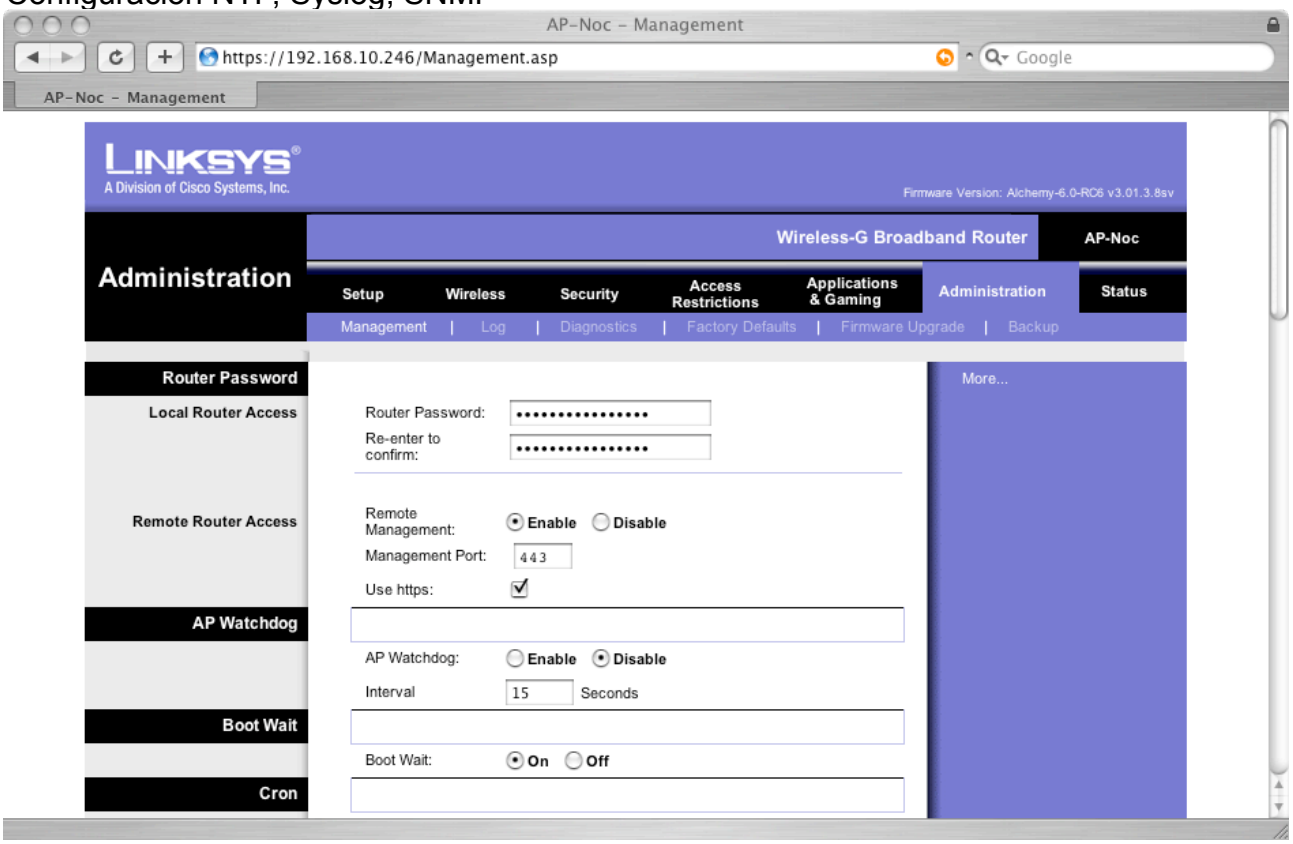
```
syslog -c /etc/syslog-ng/syslog-ng-conf
```

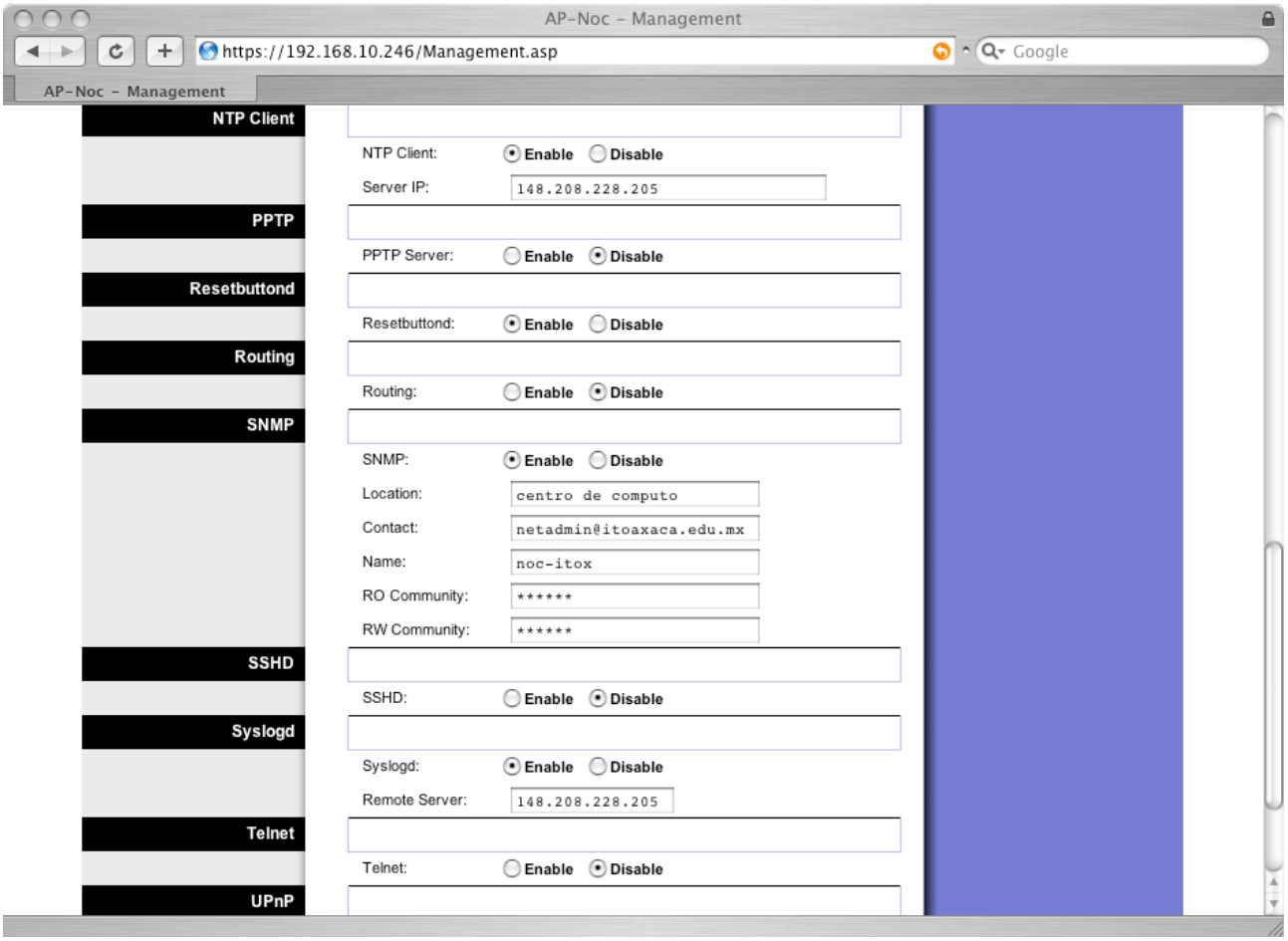
Configuración del AP Linksys

Configuración del 802.1x



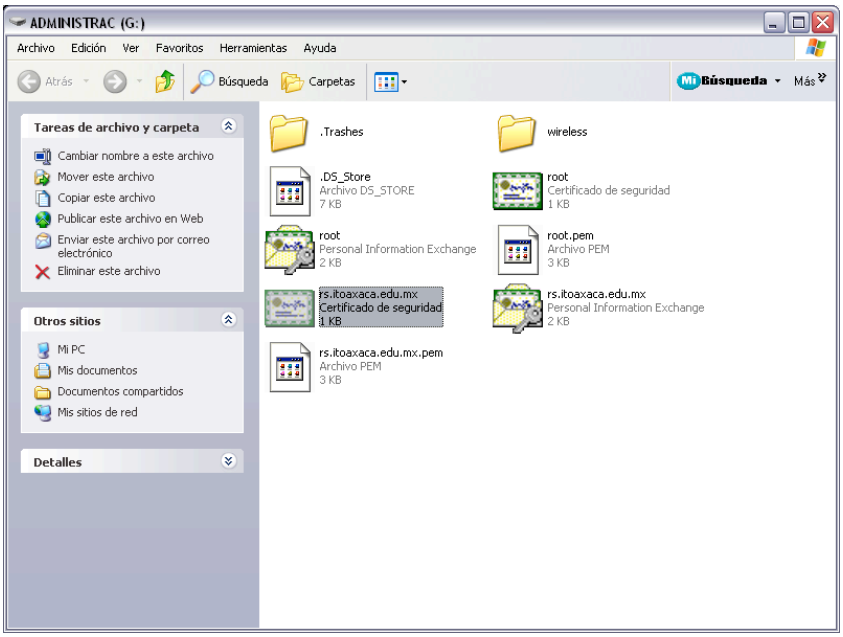
Configuración NTP, Syslog, SNMP



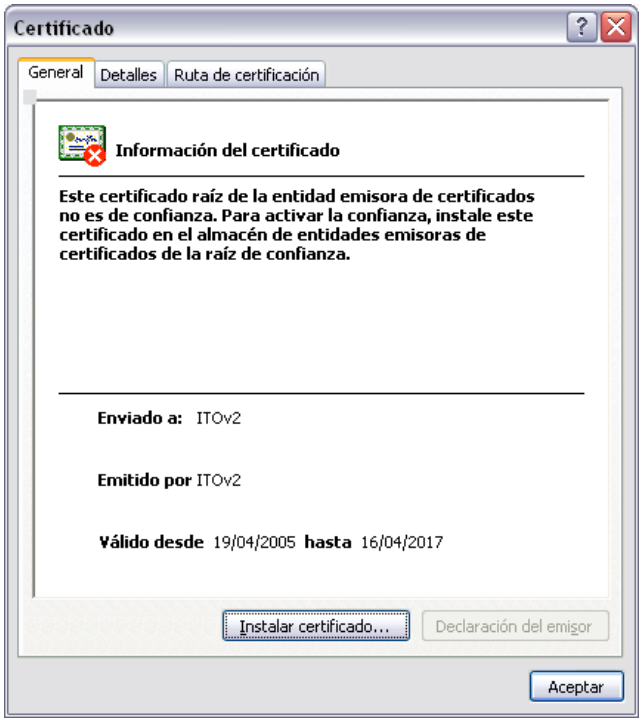

AP-Noc - Management
https://192.168.10.246/Management.asp
Google
AP-Noc - Management
NTP Client
NTP Client: Enable Disable
Server IP: 148.208.228.205
PPTP
PPTP Server: Enable Disable
Resetbuttond
Resetbuttond: Enable Disable
Routing
Routing: Enable Disable
SNMP
SNMP: Enable Disable
Location: centro de computo
Contact: netadmin@itoaxaca.edu.mx
Name: noc-itox
RO Community: ******
RW Community: ******
SSHD
SSHD: Enable Disable
Syslogd
Syslogd: Enable Disable
Remote Server: 148.208.228.205
Telnet
Telnet: Enable Disable
UPnP

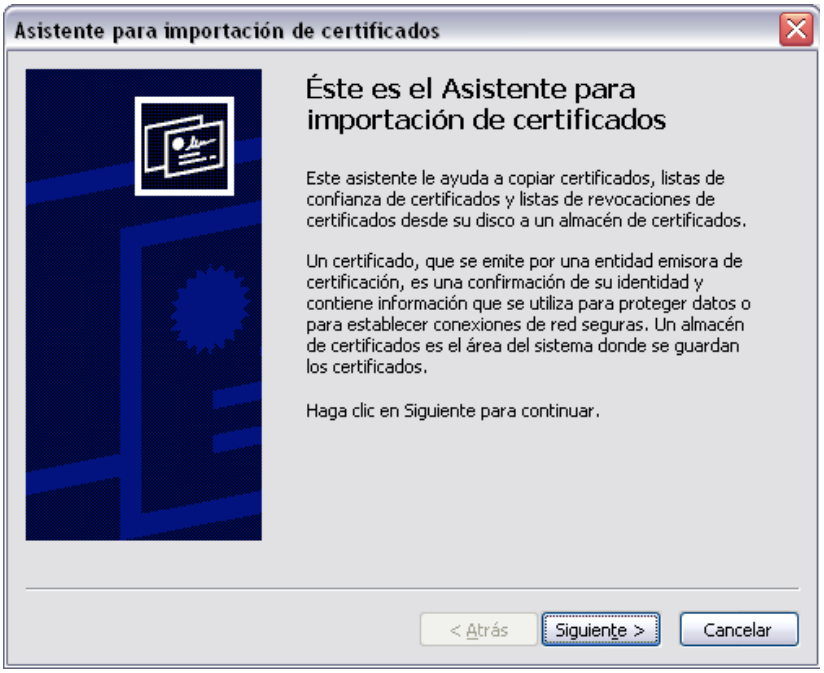
Configuración de los clientes Win XP

1. - Tener los certificados

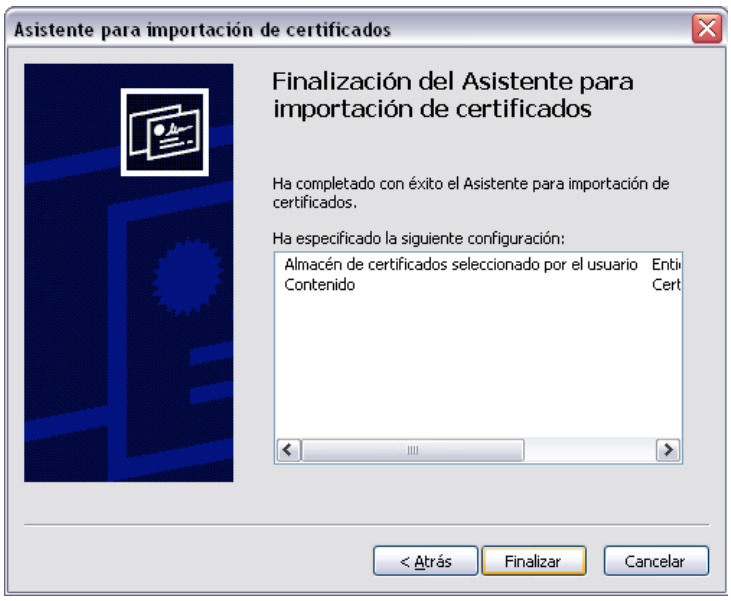


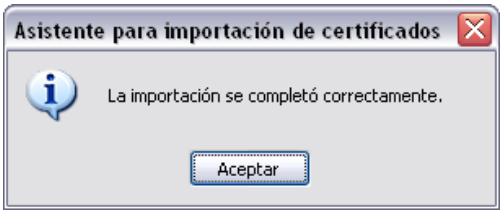
2.- Instalar los certificados empezando por el root (CA)



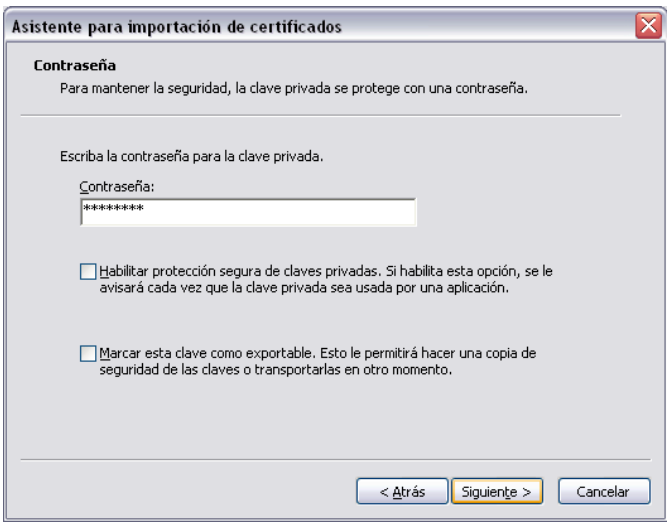

Asistente para importación de certificados
Éste es el Asistente para importación de certificados
Este asistente le ayuda a copiar certificados, listas de confianza de certificados y listas de revocaciones de certificados desde su disco a un almacén de certificados.
Un certificado, que se emite por una entidad emisora de certificación, es una confirmación de su identidad y contiene información que se utiliza para proteger datos o para establecer conexiones de red seguras. Un almacén de certificados es el área del sistema donde se guardan los certificados.
Haga clic en Siguiente para continuar.
< Atrás
Siguiente >
Cancelar


Asistente para importación de certificados
Almacén de certificados
Los almacenes de certificados son áreas del sistema donde se guardan los certificados.
Windows puede seleccionar automáticamente un almacén de certificados, o bien es posible especificar una ubicación para el certificado.
Seleccionar automáticamente el almacén de certificados en base al tipo de certificado
Colocar todos los certificados en el siguiente almacén
Almacén de certificados:
Entidades emisoras raíz de confianza
Examinar...
< Atrás
Siguiente >
Cancelar

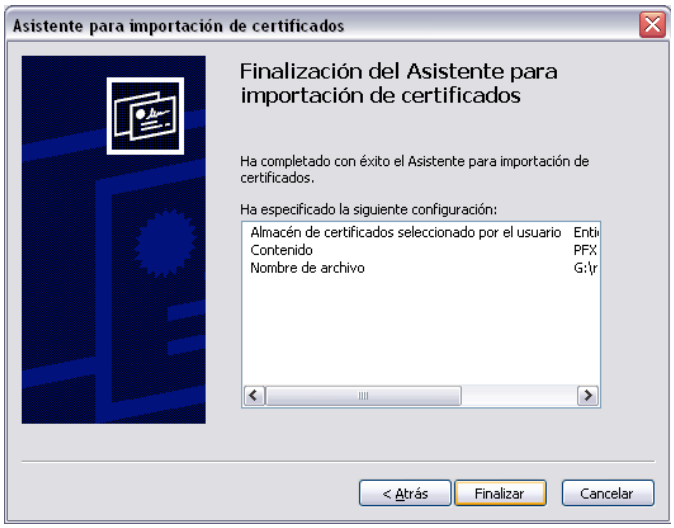


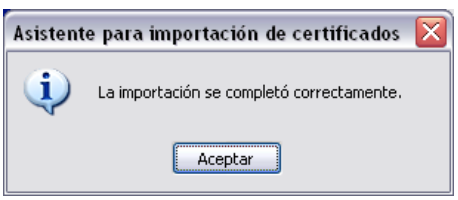


3.- Instalar el certificado del servidor rs.itoaxaca.edu.mx

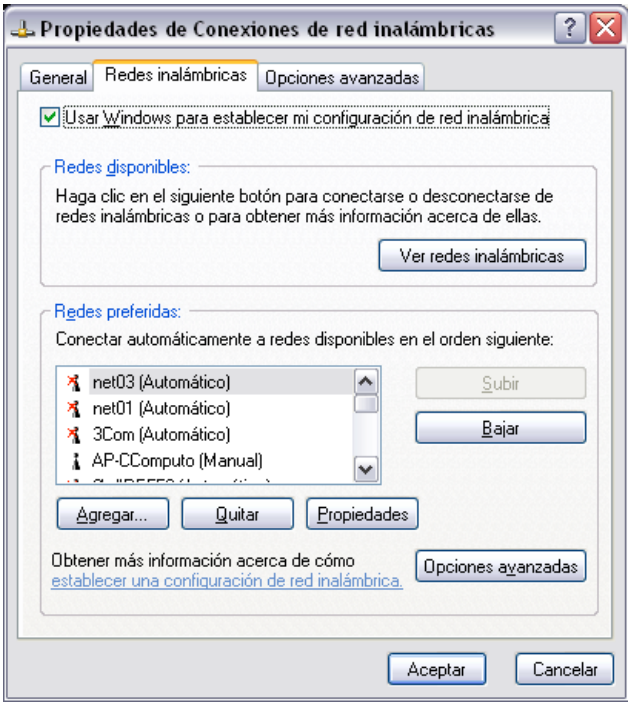

Asistente para importación de certificados
Contraseña
Para mantener la seguridad, la clave privada se protege con una contraseña.
Escriba la contraseña para la clave privada.
Contraseña:
****
Habilitar protección segura de claves privadas. Si habilita esta opción, se le avisará cada vez que la clave privada sea usada por una aplicación.
Marcar esta clave como exportable. Esto le permitirá hacer una copia de seguridad de las claves o transportarlas en otro momento.
< Atrás
Siguiente >
Cancelar

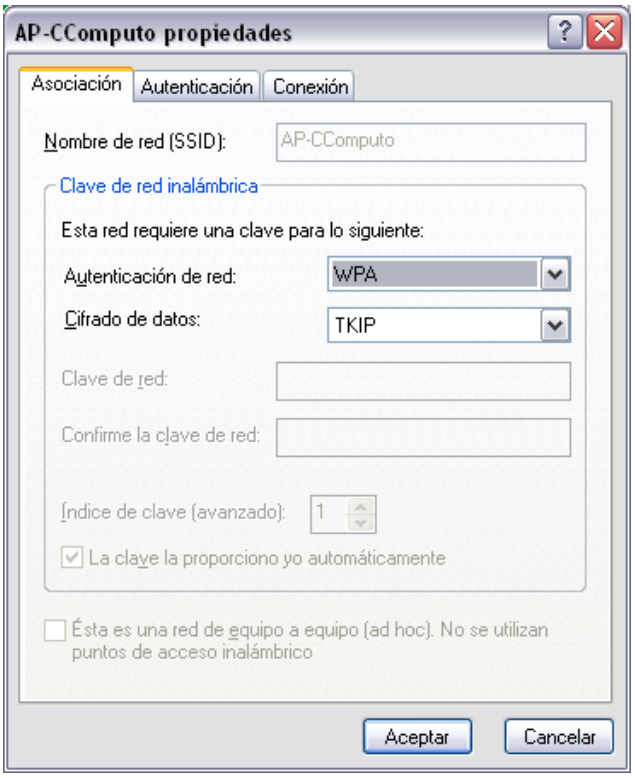

Asistente para importación de certificados
Almacén de certificados
Los almacenes de certificados son áreas del sistema donde se guardan los certificados.
Windows puede seleccionar automáticamente un almacén de certificados, o bien es posible especificar una ubicación para el certificado.
Seleccionar automáticamente el almacén de certificados en base al tipo de certificado
Colocar todos los certificados en el siguiente almacén
Almacén de certificados:
Entidades emisoras raíz de confianza
Examinar...
< Atrás
Siguiente >
Cancelar

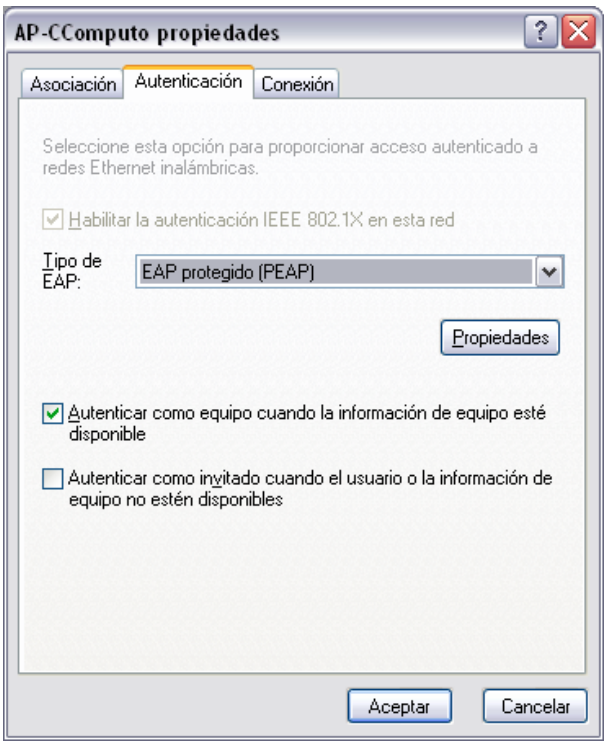

Asistente para importación de certificados
Finalización del Asistente para importación de certificados
Ha completado con éxito el Asistente para importación de certificados.
Ha especificado la siguiente configuración:
Almacén de certificados seleccionado por el usuario
Contenido
PFX
Nombre de archivo
G:\r
< Atrás
Finalizar
Cancelar

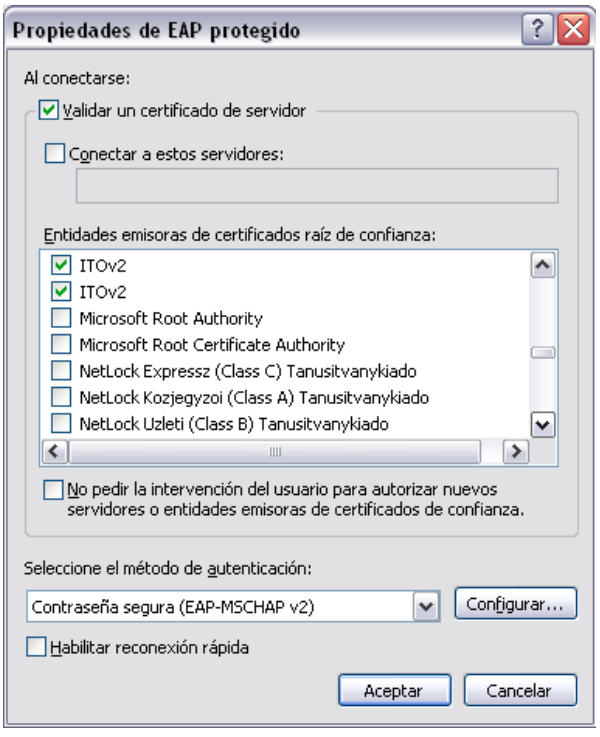

Asistente para importación de certificados
La importación se completó correctamente.
Aceptar

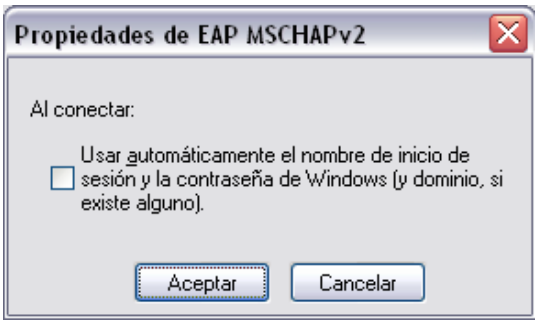
4.- Configuración de la tarjeta






AP-CComputo propiedades
Asociación
Autenticación
Conexión
Seleccione esta opción para proporcionar acceso autenticado a redes Ethernet inalámbricas.
Habilitar la autenticación IEEE 802.1X en esta red
Tipo de EAP:
EAP protegido (PEAP)
Propiedades
Autenticar como equipo cuando la información de equipo esté disponible
Autenticar como invitado cuando el usuario o la información de equipo no estén disponibles
Aceptar
Cancelar


Propiedades de EAP protegido
Al conectarse:
Validar un certificado de servidor
Conectar a estos servidores:
Entidades emisoras de certificados raíz de confianza:
ITOv2
ITOv2
Microsoft Root Authority
Microsoft Root Certificate Authority
NetLock Expressz (Class C) Tanusitvanykiado
NetLock Kozjegyzoi (Class A) Tanusitvanykiado
NetLock Uzleti (Class B) Tanusitvanykiado
No pedir la intervención del usuario para autorizar nuevos servidores o entidades emisoras de certificados de confianza.
Seleccione el método de autenticación:
Contraseña segura (EAP-MSCHAP v2)
Configurar...
Habilitar reconexión rápida
Aceptar
Cancelar


Propiedades de EAP MSCHAPv2
Al conectar:
Usar automáticamente el nombre de inicio de sesión y la contraseña de Windows (y dominio, si existe alguno).
Aceptar
Cancelar