

## 1. Radius con DNIE.

FreeRadius es un paquete de software de código abierto y libre distribución que permite implementar un servidor de RADIUS. El servidor de FreeRadius es modular, para facilitar su extensión, y es muy escalable. Además el almacenamiento de la información de autenticación (usuarios/contraseña) se puede realizar directamente (sobre ficheros de textos de configuración propios) o bien preguntando a bases de datos externas, como MySQL o bien ficheros del sistema/etc/passwd

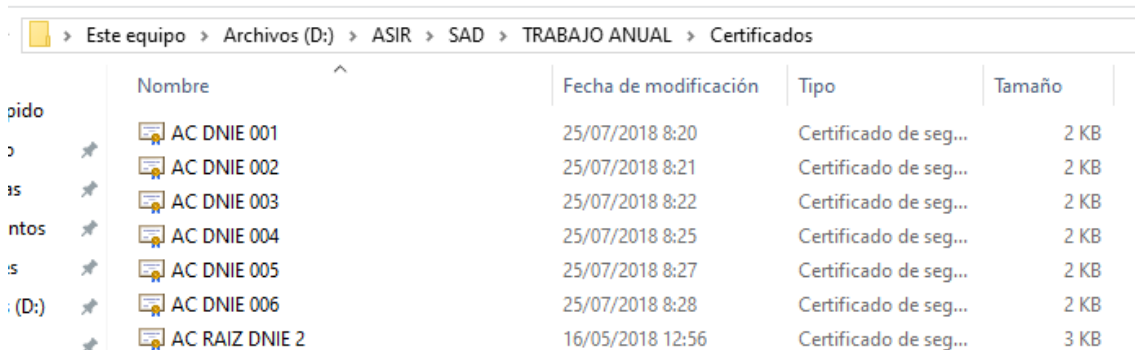
### a. PKI del DNIE

Las Autoridades de Certificación que la componen son:

- Una **Autoridad de Certificación raíz** que sólo emite certificados para sí misma y sus Autoridades de Certificación subordinadas.
- Tres **Autoridades de Certificación subordinadas** que emiten certificados para los titulares de DNIE, a cada ciudadano le corresponde uno de ellos, como a priori no sabemos cuál será el que firme el certificado de un ciudadano concreto, lo que haremos es concatenar los tres certificados en un fichero junto a la Autoridad raíz de manera que FreeRADIUS pedirá a OpenSSL que recorra el fichero buscando el certificado necesario para un cliente dado siempre que dejemos el certificado de la Autoridad raíz en último lugar.

Primero hay que descargar los cuatro certificados de la web del DNIE:

[https://www.dnielectronico.es/PortalDNIE/PRF1\\_Cons02.action?pag=REF\\_076&id\\_menu=68](https://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_076&id_menu=68)



	Nombre	Fecha de modificación	Tipo	Tamaño
pedido	AC DNIE 001	25/07/2018 8:20	Certificado de seg...	2 KB
o	AC DNIE 002	25/07/2018 8:21	Certificado de seg...	2 KB
is	AC DNIE 003	25/07/2018 8:22	Certificado de seg...	2 KB
ntos	AC DNIE 004	25/07/2018 8:25	Certificado de seg...	2 KB
is	AC DNIE 005	25/07/2018 8:27	Certificado de seg...	2 KB
(D:)	AC DNIE 006	25/07/2018 8:28	Certificado de seg...	2 KB
	AC RAIZ DNIE 2	16/05/2018 12:56	Certificado de seg...	3 KB

Los descomprimos y pasamos los cuatro ficheros por SFTP al servidor.

```
root@Proxy:/home/frodo/Certificate# ls
ACDNIE001-SHA1.crt ACDNIE002-SHA1.crt ACDNIE003-SHA1.crt ACRAIZ-SHA1.cer
```

Los ficheros vienen en formato DER (.crt), un formato propio de Windows, lo pasamos a PEM que es un formato más propio de Linux para que no haya ningún problema, lo haremos con OpenSSL situándonos en el directorio donde hemos copiado los ficheros ejecutamos lo siguiente.

```
root@Proxy:/home/frodo/Certificate# rm ACDNIE001.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 001.crt -out ACDNIE001.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 002.crt -out ACDNIE002.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 003.crt -out ACDNIE003.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 004.crt -out ACDNIE004.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 005.crt -out ACDNIE005.pem
root@Proxy:/home/frodo/Certificate# openssl x509 -inform DER -outform PEM -in AC\ DNIE\ 006.crt -out ACDNIE006.pem
root@Proxy:/home/frodo/Certificate# ls
ACDNIE001.pem ACDNIE002.pem ACDNIE003.pem ACDNIE004.pem ACDNIE005.pem ACDNIE006.pem AC DNIE 001.crt AC DNIE 002.crt AC DNIE 003.crt AC DNIE 004.crt AC DNIE 005.crt AC DNIE 006.crt AC RAIZ DNIE 2.crt
root@Proxy:/home/frodo/Certificate#
```

Ahora toca concatenar los certificados en único archivo que llamaremos "todas.pem".

```

root@Proxy:/home/frodo/Certificate# cp AC\ RAIZ\ DNIE\ 2.crt todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI001.pem >> todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI002.pem >> todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI003.pem >> todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI004.pem >> todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI005.pem >> todas.pem
root@Proxy:/home/frodo/Certificate# cat ACDNI006.pem >> todas.pem

```

## b. Instalación de FreeRADIUS

Descargaremos FreeRadius 3.0.20 y lo instalaremos.

```

root@Proxy:/home/frodo# wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-3.0.20.tar.gz

```

```

root@Proxy:/home/frodo# tar xzf freeradius-server-3.0.20.tar.gz

```

Será necesario instalar los siguientes paquetes porque contienen librerías de desarrollo necesarias para la instalación de FreeRADIUS.

```

root@Proxy:/home/frodo# cd freeradius-server-3.0.20
root@Proxy:/home/frodo/freeradius-server-3.0.20# apt-get install libtalloc-dev

```

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# apt install libssl-dev

```

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# apt install build-essential

```

Ejecutamos él. /configure.

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# ./configure

```

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# make

```

Instalamos radius.

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# make install

```

## c. Configuración de FreeRADIUS

Por defecto la instalación se realiza en /usr/local/etc/raddb/.

Empezaremos con el fichero radius.conf, en el apartado PROXY CONFIGURATION deshabilitaremos el servidor proxy y comentaremos el include, no es estrictamente necesario pero ahorrará recursos.

```

root@Proxy:/home/frodo/freeradius-server-3.0.20# nano /usr/local/etc/raddb/radiusd.conf

```

```

proxy_requests = no
$INCLUDE proxy.conf

```

En el apartado *security* modificamos el parámetro *allow\_vulnerable\_openssl* para permitir que FreeRADIUS se inicie con una versión vulnerable de OpenSSL.

```
#  
allow_vulnerable_openssl = yes
```

Continuaremos con el fichero `clients.conf`, aquí añadimos el o los clientes RADIUS esto es el punto de acceso inalámbrico o en mi caso el router inalámbrico que configuraremos con los datos aquí introducidos, añadimos el cliente al final del fichero con la IP y prefijo que tendrá el punto de acceso inalámbrico, la `secret` es una clave cualquiera que permitirá comunicarse el punto de acceso con el servidor, y un nombre para el cliente que estamos introduciendo (`shortname`).

```
root@Proxy:/home/frodo# nano /usr/local/etc/raddb/clients.conf
```

```
1  client S&SFree {  
2      ipaddr      = 192.168.3.1  
3      secret      = bolson  
4      shostname    = S&SFree  
5  }
```

Continuamos con el archivo `default` dentro de `sites-enabled`, en este fichero se configuran los distintos host virtuales al igual que haríamos con Apache. En las secciones `listen` no es estrictamente necesario modificar nada, el servidor funcionará igualmente con la configuración que viene por defecto, las podemos configurar de la siguiente manera para hacer el servidor más seguro evitando que escuche peticiones provenientes de una IP o puertos distintos a los que tenemos planeado.

La primera sección `listen` (paquetes de autenticación) la configuramos de la siguiente manera, siendo `ipv4addr` la IP del adaptador que está conectado al mikrotik.

```
root@Proxy:/home/frodo# nano /usr/local/etc/raddb/sites-enabled/default
```

```
type = auth
```

```
ipv4addr = 192.168.3.2
```

```
# Port on which to listen.
```

```
# Allowed values are:
```

```
# integer port number (1812)
```

```
# 0 means "use /etc/services for the proper port"
```

```
port = 1812
```

La segunda sección `listen` (paquetes de contabilización) la configuramos de la siguiente manera. La opción `ipaddr` se deberá cambiar por `ipv4addr`.

```
#
listen {
    ipv4addr = 192.168.3.2
    # ipv6addr = ::
    port = 1813
    type = acct
    # interface = eth0
    # clients = per_socket_clients
```

En la sección *authorize* tenemos que deshabilitar el filtro de nombre de usuario, éste comprueba si el nombre de usuario contiene espacios o caracteres inválidos rechazando el acceso en su caso. El CommonName del certificado de cliente contenido en el DNle se compone de apellidos, nombre, espacios en blanco, una coma y el literal “(AUTENTICACIÓN)”, el filtro rechazaría el acceso no permitiendo que continúe la petición del cliente.

```
#filter_username
```

Como nuestra intención es que los usuarios únicamente puedan tener acceso a la red WiFi mediante el DNle, deshabilitaremos los módulos de autenticación que no vamos a utilizar, dejaremos *eap* habilitado ya que es la manera en que se autenticaran los clientes.

```
#chap
```

```
#mschap
```

```
#digest
```

```
#suffix
```

```
#pap
```

Una vez que el usuario está autorizado la sección *authenticate* se encargara de instanciar el módulo de autenticación, esta sección la podemos dejar tal cual ya que hemos descartado los módulos que no vamos a utilizar en la sección anterior.

Por ultimo debemos en el módulo *eap* tenemos que indicar dónde está el certificado de autoridad que controlará las credenciales de los clientes y firmará sus certificados contenidos en el DNle.

```
root@Proxy:/home/frodo# nano /usr/local/etc/raddb/mods-enabled/eap
```

```
ca_file = /usr/local/etc/raddb/certs/todas.pem
```

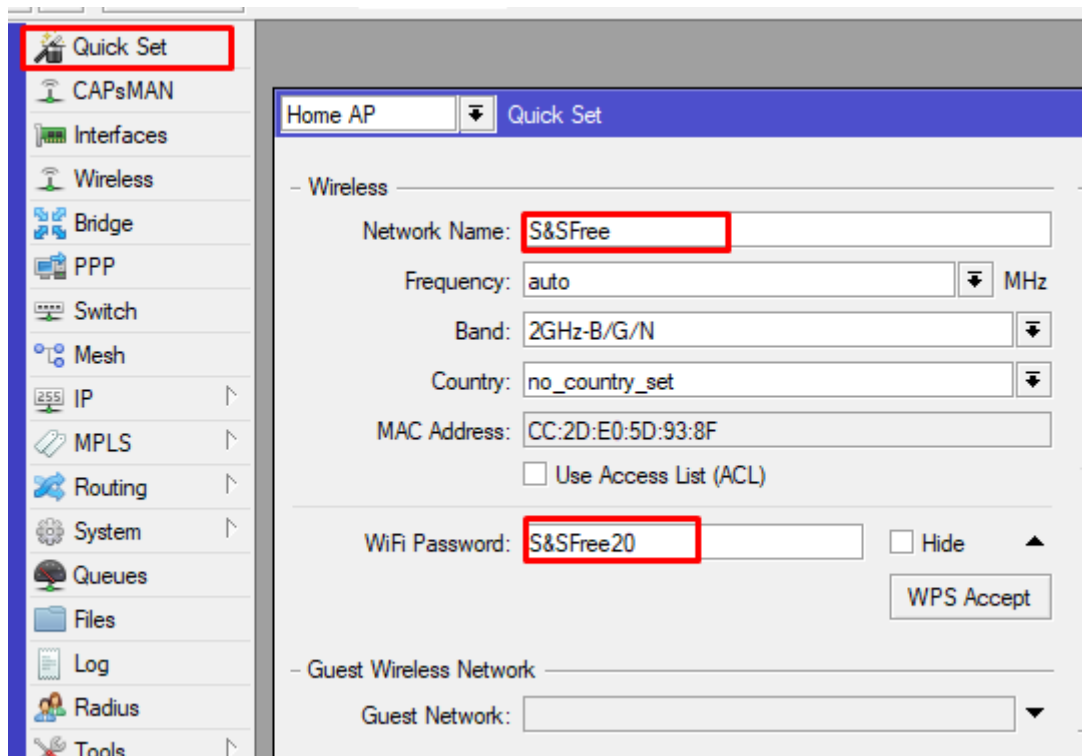
Por ultimo copiamos el fichero con los certificados al directorio de certificados de FreeRADIUS.

```
root@Proxy:/home/frodo# cp Certificate/todas.pem /usr/local/etc/raddb/certs/
```

#### d. Punto de acceso

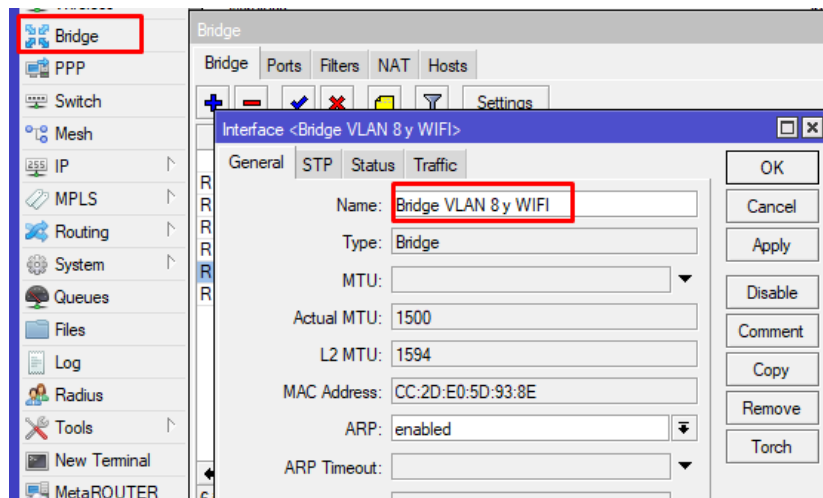
Como cliente RADIUS hace falta configurar un punto de acceso inalámbrico, en este caso se hará con un mikrotik. Lo configuramos con los siguientes pasos.

**Pondremos un nombre a la WiFi, en la sección *Quick Set*.**

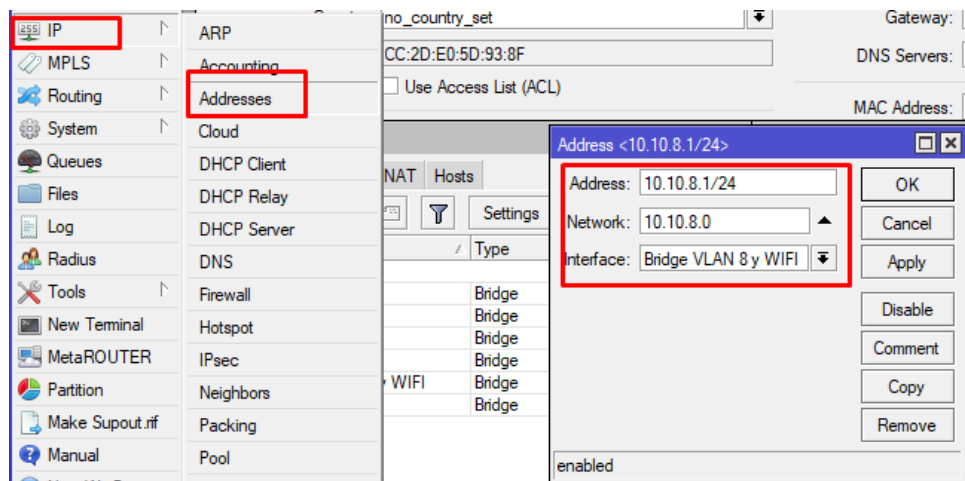


Crearemos un Bridge para la WLAN y definiremos una Ip para dicho Bridge.

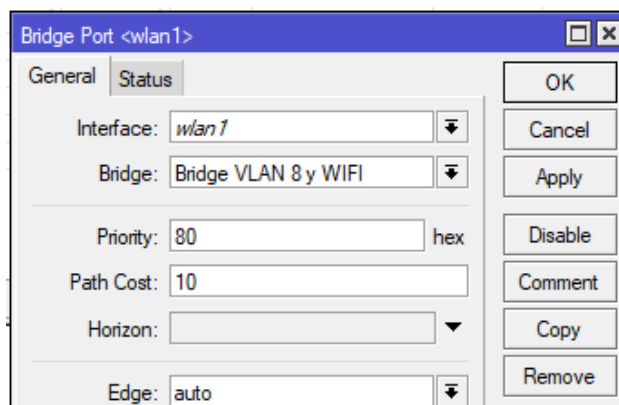
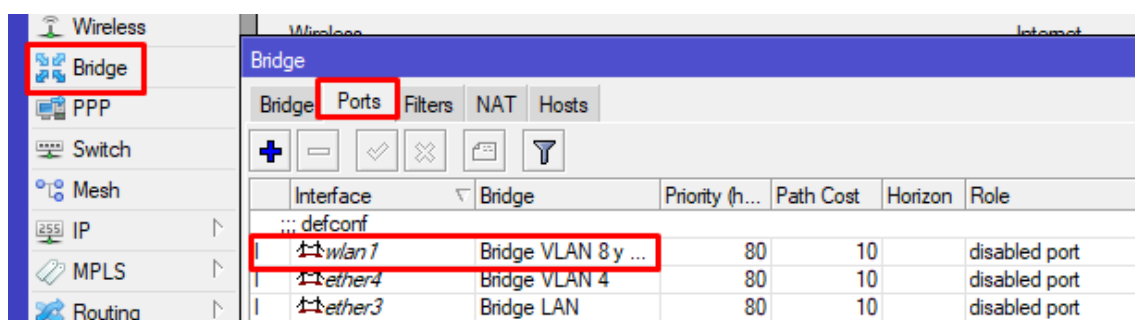
Bridge:



Definir una ip para ese bridge.



Lo siguiente que haremos será asociar el bridge que hemos creado a la interfaz de wlan.



En este caso también se ha creado un servidor DHCP para que reparta IP por la Wifi, pero eso es irrelevante para hacer esta práctica.

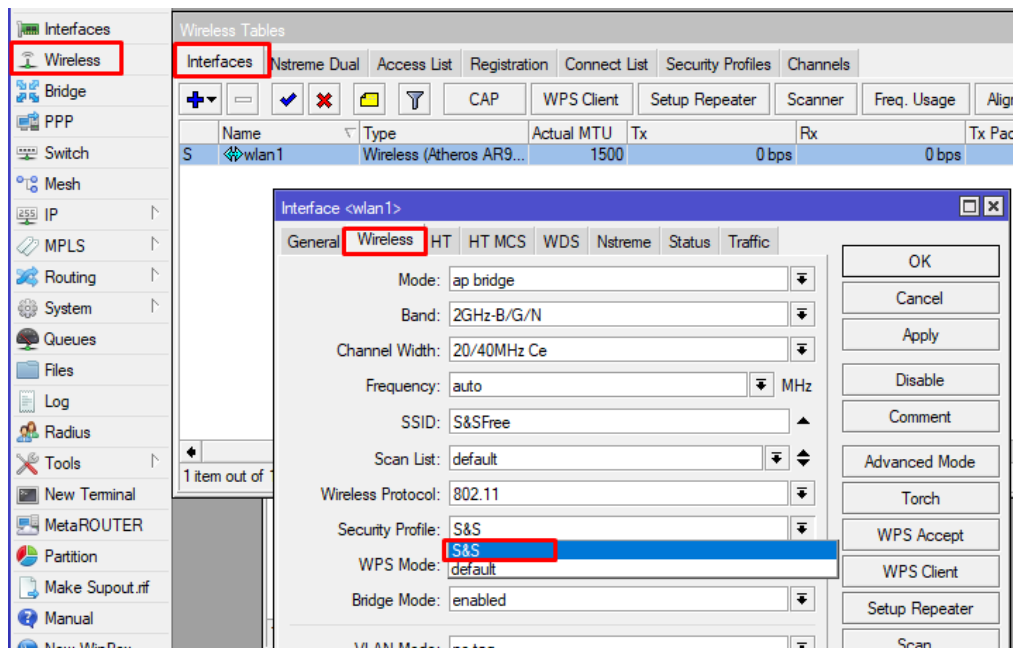
Una vez configurado esto nos iremos a la sección de Radius para hacer la comunicación con el servidor.

En esta sección lo que haremos será marcar las opciones de hotspot y Wireless, pondremos la IP del servidor Radius, la contraseña que hemos definido en el archivo client.conf y los puertos del servidor Radius.

Después de establecer conexión con el servidor lo que haremos será ir a la sección de *Wireless* aquí nos iremos a la Pestaña de *Security Profiles*.

Aquí definiremos el Perfil de seguridad, le pondremos un nombre y marcaremos las opciones de WPA EAP y WPA2EAP, estas dos opciones nos dejarán autenticarnos con los DNIE.

Por ultimo deberemos de ir a la pestaña de Interfaces y seleccionar el perfil de seguridad que acabamos de crear.

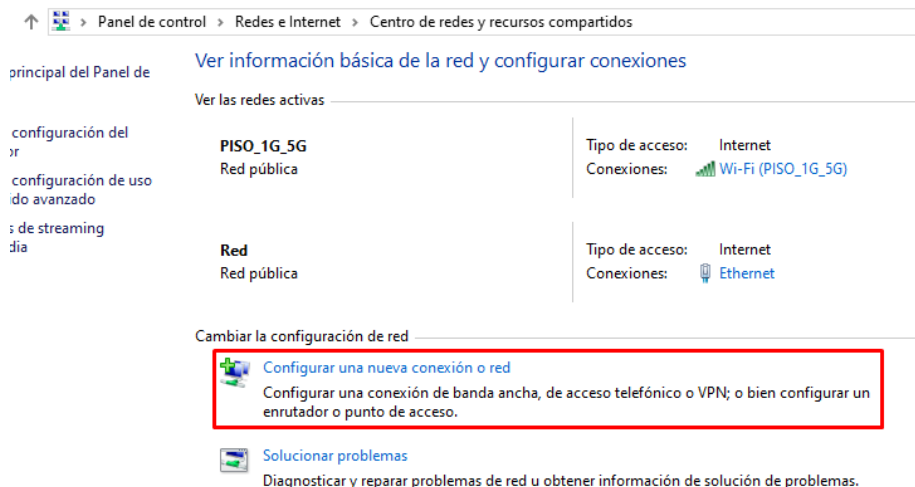


## e. Clientes

Para que un cliente pueda conectarse a la red WiFi será necesario un PC o portátil con tarjeta de red inalámbrica, sistema operativo Windows, un lector de tarjetas inteligentes, su DNI electrónico, y según qué casos hará falta el software que proporciona el Cuerpo Nacional de Policía llamado “Módulo criptográfico para el DNLe”.

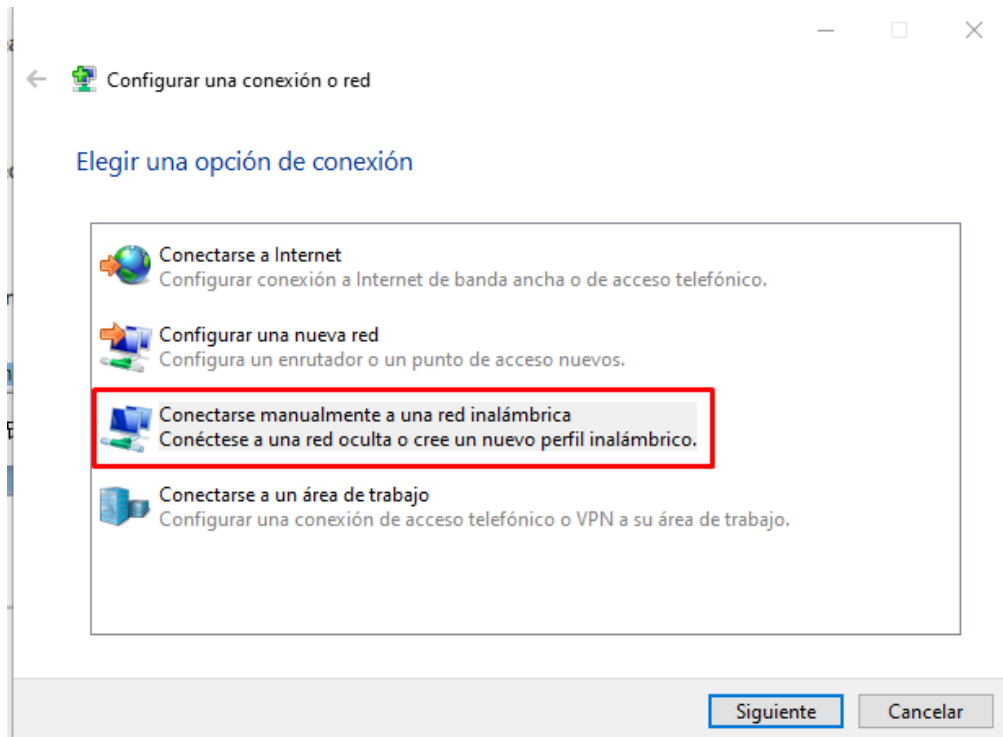
En Windows 7,8 y 10 no es necesaria su instalación ya que al introducir el DNLe en el lector automáticamente se descargarán los drivers necesarios desde Windows Update, permitiendo trabajar al DNLe como dispositivo plug&play.

Vamos a crear una nueva conexión de red inalámbrica, nos vamos a “Panel de control\Redes e Internet\Centro de redes y recursos compartidos”, pinchamos sobre “Configurar una nueva conexión o red”.



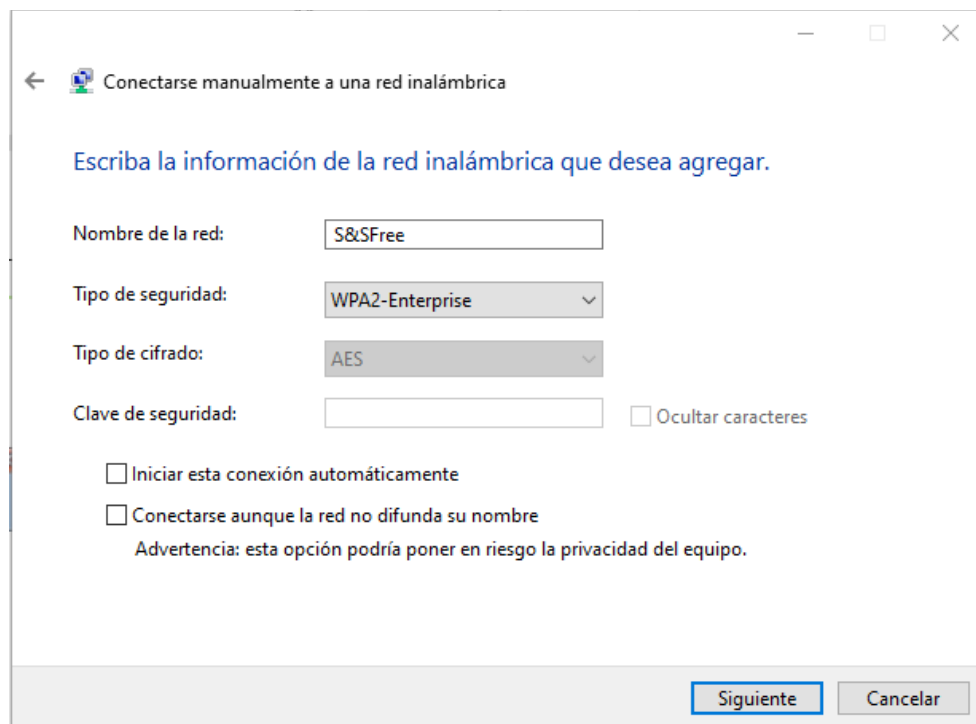


Seleccionamos la opción de “Conectarse manualmente a una red inalámbrica” y pulsamos “Siguiente”.

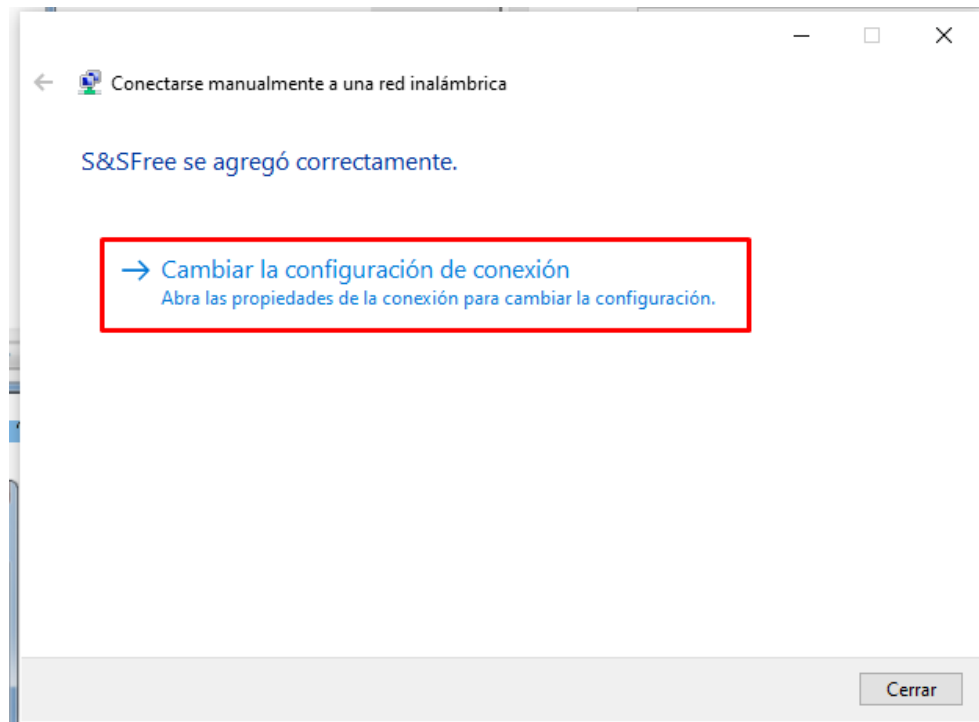


Introducimos los datos que habíamos configurado en el punto de acceso WiFi:

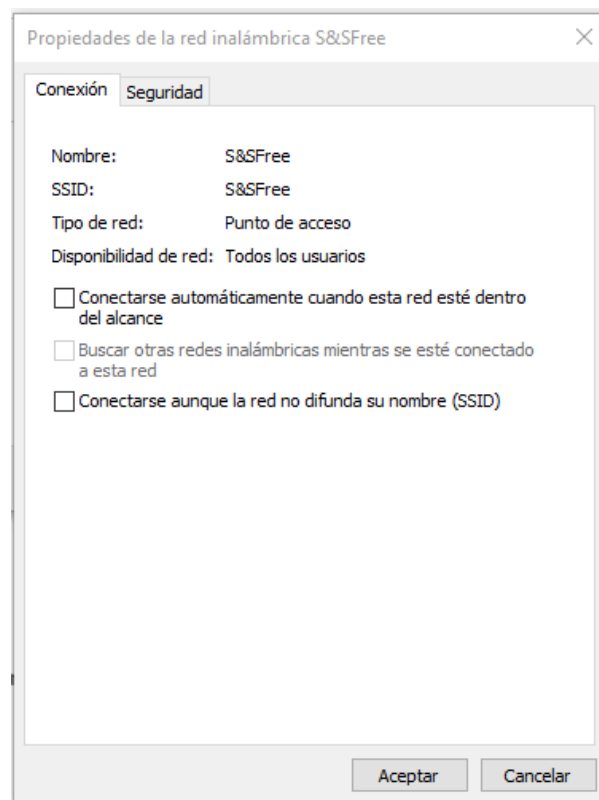
- Nombre de la red: S&SFree
- Tipo de seguridad: WPA2-Enterprise
- Tipo de cifrado: AES



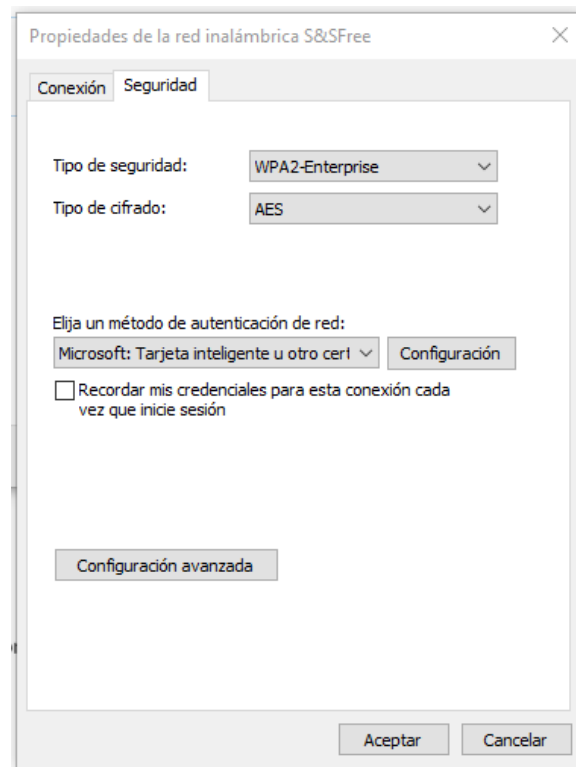
Pulsamos “Siguiente” lo que nos lleva a la siguiente ventana donde pincharemos sobre “Cambiar la configuración de conexión” para realizar algunos ajustes sobre la nueva conexión.



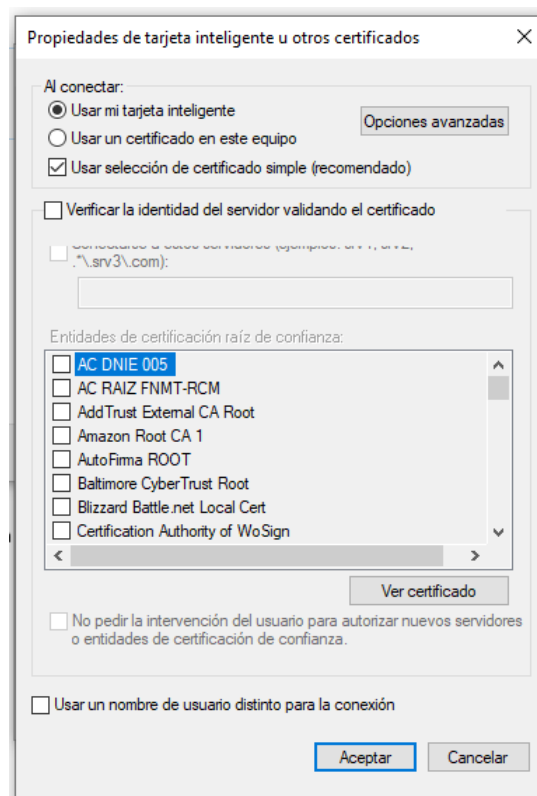
Nos aparece una nueva ventana con las propiedades de la conexión, de la pestaña “Conexión” no tocaremos nada.



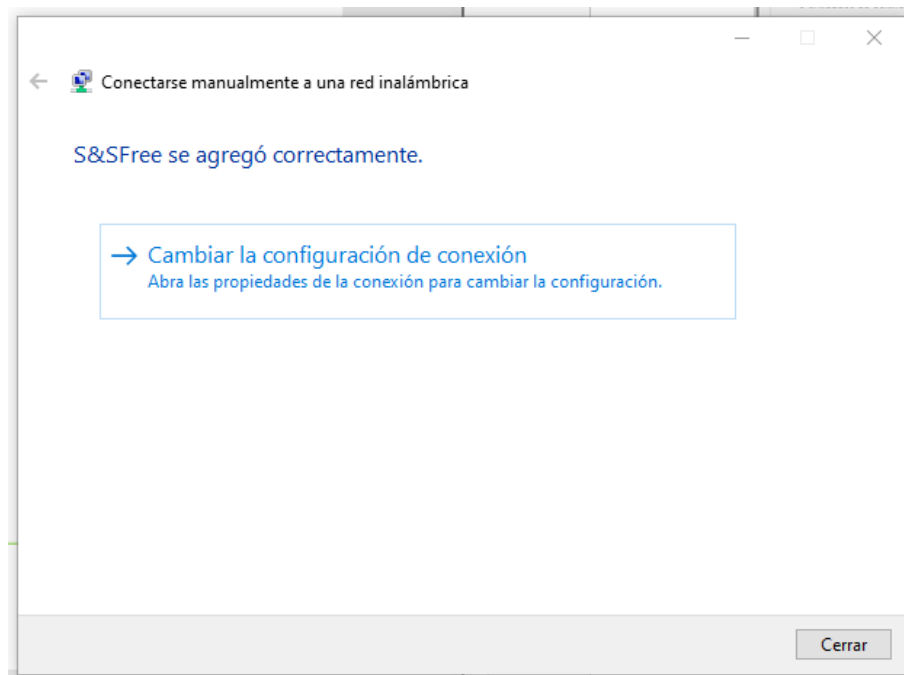
Pasamos a la pestaña “Seguridad”, en el menú “Elija un método de autenticación de red:” seleccionamos “Microsoft: Tarjeta inteligente u otro certificado”, luego pinchamos sobre el botón “Configuración”.



En “Al conectar:” seleccionamos “Usar mi tarjeta inteligente”, desmarcamos la casilla “Validar un certificado de servidor”.



Finalmente pinchamos sobre “Cerrar” en la ventana anterior con lo que tendremos configurada la conexión al servidor RADIUS mediante DNle.



## f. Pruebas

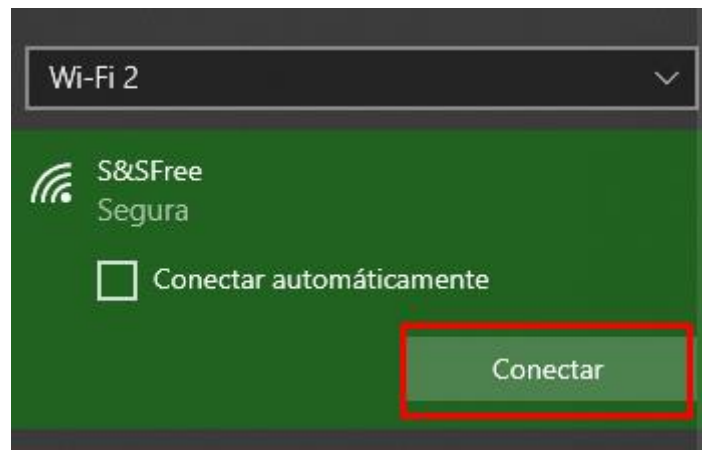
Iniciamos FreeRADIUS con alguno de los siguientes comandos, el “-X” es para iniciarlo en modo debug.

```
root@Proxy:/home/frodo# radiusd -X
```

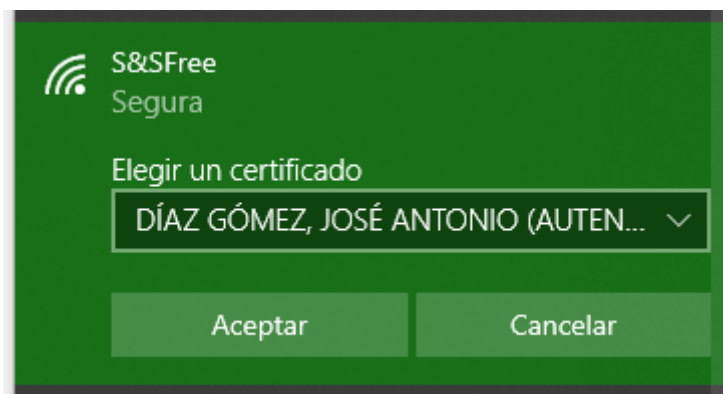
Se iniciará el servidor. Si todo ha ido bien al final del log nos aparecerá el mensaje “Ready to process requests”.

```
    }  
  }  
Listening on auth address 192.168.3.2 port 1812 bound to server default  
Listening on acct address 192.168.3.2 port 1813 bound to server default  
Ready to process requests  
||
```

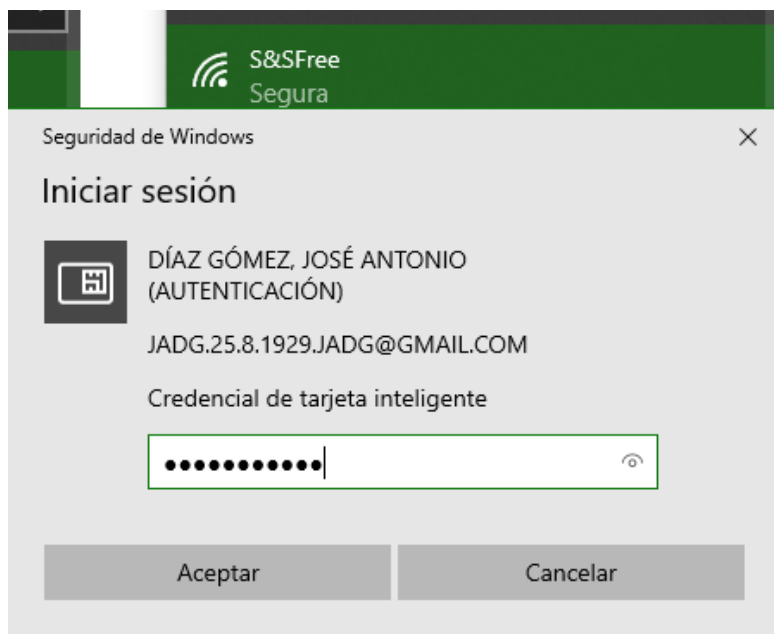
Con el lector de tarjetas inteligentes instalado, conectado, y el DNle insertado, buscamos la red WiFi que hemos configurado en el cliente FreeRADIUS y le damos a conectar.



Nos debería de seleccionar el método de autenticación de los certificados del DNle.



Seguidamente deberemos de poner el PIN del Dnie electrónico y si todo ha ido bien debería de conectar a la red.



```
(8) Received Access-Request Id 19 from 192.168.3.1:37797 to 192.168.3.2:1812 length 268
(8) Service-Type = Framed-User
(8) Framed-MTU = 1400
(8) User-Name = "DÍAZ GÓMEZ, JOSÉ ANTONIO (AUTENTICACIÓN)"
(8) State = 0xfc077bddfb0e765669e81f09fe7738e5
(8) NAS-Port-Id = "wlan1"
(8) NAS-Port-Type = Wireless-802.11
(8) Acct-Session-Id = "82200000"
(8) Acct-Multi-Session-Id = "CC-2D-E0-5D-93-8F-50-3E-AA-77-5F-DD-82-20-00-00-00-00-00"
(8) Calling-Station-Id = "50-3E-AA-77-5F-DD"
(8) Called-Station-Id = "CC-2D-E0-5D-93-8F:S&SFree"
(8) EAP-Message = 0x020900060d00
(8) Message-Authenticator = 0xfb2291f8dd389ac2090ad844ce1f4fd2
(8) NAS-Identifier = "MikroTik"
```

Para poder definir que solo los usuarios que estén en el sistema se puedan loguear se tendrá que definir en el fichero **users** las siguientes opciones.

```
root@Proxy:/home/frodo# nano /usr/local/etc/raddb/users
```

```
# be given any additional resources.
#
DEFAULT Auth-Type := Reject
|       Reply-Message = "ACCESO DENEGADO."
#
```

Ahora comprobaremos que no nos deja loguearnos y nos saldrá el siguiente mensaje de error.

```
files: users: Matched entry DEFAULT at line 66
[files] = ok
[expiration] = noop
[logintime] = noop
} # authorize = updated
Found Auth-Type = Reject
Auth-Type = Reject, rejecting user
Failed to authenticate the user
Using Post-Auth-Type Reject
```

Para poder loguearnos con un usuario específico solo tendremos que poner las siguientes líneas en el fichero **users**. En este fichero pondré el nombre del usuario y el tipo de protocolo que será **EAP**. La opción definida anteriormente se dejará sin comentar para que solo los usuarios que estén definidos en dicho archivo puedan acceder a la Wi-Fi.

```
"DÍAZ GÓMEZ, JOSÉ ANTONIO (AUTENTICACIÓN)" Auth-Type := EAP
|       Reply-Message = "Acceso permitido para el usuario: %{User-Name}."
```

```
#
# Deny access for a group of users.
#
# Note that there is NO 'Fall-Through' attribute, so the user will not
# be given any additional resources.
#
DEFAULT Auth-Type := Reject
|       Reply-Message = "ACCESO DENEGADO."
```

Ahora comprobaremos que el usuario se loguea en la red Wi-Fi y sale el mensaje que hemos definido en el fichero.

```
(8) eap: Peer sent EAP Response (code 2) ID 9 length 6
(8) eap: No EAP Start, assuming it's an on-going EAP conversation
(8) [eap] = updated
(8) files: users: Matched entry DÍAZ GÓMEZ, JOSÉ ANTONIO (AUTENTICACIÓN) at line 57
(8) files: EXPAND Acceso permitido para el usuario: %{User-Name}.
(8) files: --> Acceso permitido para el usuario: DÍAZ GÓMEZ, JOSÉ ANTONIO (AUTENTICACIÓN).
(8) [files] = ok
```

Vemos como se ha conectado a nuestra red.



Comprobamos que utilizando un DNle que no está en el fichero **users** definido no es posible conectarnos a la red Wi-Fi.

```
} # authorize = updated
Found Auth-Type = Reject
Auth-Type = Reject, rejecting user
Failed to authenticate the user
Using Post-Auth-Type Reject
# Executing group from file /usr/local/etc/raddb/sites-enabled/default
Post-Auth-Type REJECT {
attr_filter.access_reject: EXPAND %{User-Name}
attr_filter.access_reject: --> DÍAZ GÓMEZ, ROCÍO (AUTENTICACIÓN)
attr_filter.access_reject: Matched entry DEFAULT at line 11
[attr_filter.access_reject] = updated
... Expanding EAP session with state 0x744d0e08744d0e08
```



## 2. WEBGRAFÍA.

### DNle

<https://drive.google.com/drive/folders/1tArImfmwXLazmNixgPJaqmIC1wIFCYJc>

[http://oa.upm.es/1602/1/PFC\\_SERGIO\\_YEBENES\\_MORENO.pdf](http://oa.upm.es/1602/1/PFC_SERGIO_YEBENES_MORENO.pdf)

<http://seguridadxredes.blogspot.com/2015/12/vpn-ipsec-ii-autenticando-con-dnie-o.html>

<https://www.eduardocollado.com/2017/07/24/conexion-a-mikrotik-via-radius/>