



## CONFIGURACIÓN DE EQUIPOS LINUX PARA LA CONEXIÓN A LA RED INALÁMBRICA eduroam



### Requisitos específicos

Suplicante WP como por ejemplo [wpa\\_supplicant](#). (En la [web](#) de este producto disponemos de una lista de tarjetas wireless soportadas y drivers a usar).

### Configuración de la conexión

- Configurar el cliente wpa\_supplicant mediante el fichero de configuración `/etc/wpa_supplicant.conf`.

Este es un ejemplo de cómo debe ser el fichero para conectar al SSID "**eduroam**":

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=1
ap_scan=1
fast_reauth=1
network={
ssid="eduroam"
key_mgmt=WPA-EAP
proto=WPA
# Si la tarjeta soporta WPA2 con AES usaremos proto=RSN
pairwise=CCMP TKIP
group=CCMP TKIP
eap=TTLS
anonymous_identity="anonymous@unizar.es"
identity="mi-login-de-correo"
password="mi-password"
phase2="auth=PAP"
}
```

Es conveniente proteger dicho archivo con los permisos adecuados de lectura ya que las credenciales se encuentran en él.

- Usuarios de otra institución distinta a la Universidad de Zaragoza adscrita al proyecto EDUROAM, deberán sustituir **mi-login-de-correo** por el **login-de-su-institución@dominio** (por ejemplo: pepe@ugr.es)
- Ahora deberemos comprobar qué driver del wpa\_supplicant sirve para el chipset/tarjeta-WiFi/driver que estemos utilizando, veamos que drivers puede utilizar el wpa\_supplicant:

```
prompt$ wpa_supplicant -h
wpa_supplicant v0.3.8
Copyright (c) 2003-2005, Jouni Malinen and contributors
```

This program is free software. You can distribute it and/or modify it under the terms of the GNU General Public License version 2.

Alternatively, this software may be distributed under the terms of the BSD license. See README and COPYING for more details.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

usage:

```
wpa_supplicant [-BddehLqqvw] -i -c [-D] [-N -i -c [-D] ...]
```

drivers:

```
hostap = Host AP driver (Intersil Prism2/2.5/3)
prism54 = Prism54.org driver (Intersil Prism GT/Duette/Indigo)
madwifi = MADWIFI 802.11 support (Atheros, etc.)
atmel = ATMEL AT76C5XXx (USB, PCMCIA)
wext = Linux wireless extensions (generic)
ndiswrapper = Linux ndiswrapper
ipw = Intel ipw2100/2200 driver
```

options:

```
-B = run daemon in the background
-d = increase debugging verbosity (-dd even more)
-K = include keys (passwords, etc.) in debug output
-t = include timestamp in debug messages
-h = show this help text
-L = show license (GPL and BSD)
-q = decrease debugging verbosity (-qq even less)
-v = show version
-w = wait for interface to be added, if needed
-N = start describing new interface
```

- Si nuestra combinación chipset/tarjeta-WiFi/driver se ajusta a alguna de las opciones, un script como el que sigue activará la conexión WiFi 802.1x:

```
ifconfig mi-interfaz-wifi up
wpa_supplicant -B -i mi-interfaz-wifi -c /etc/wpa_supplicant.conf -dd -D
mi-driver
sleep 5
dhclient mi-interfaz-wifi
```

- Por ejemplo en el caso de un Intel Centrino con tarjeta Intel Pro Wireless 2100, la línea que invoca al wpa\_supplicant podrá sustituirse por:

```
wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf -D ipw
```

(las opciones -dd son para depuración y si todo va bien no harán falta).

En caso de tener problemas con wpa\_supplicant se puede usar otro suplicante como [Xsupplicant](#)

Instrucciones linux basadas en el manual de la [Universidad Carlos III de Madrid](#)