

Using wpa_supplicant to Manage your Wireless Network Connections#

The *wpa_supplicant* daemon is the "standard" mechanism used to provide persistence of wireless networking information as well as managing automated connections into networks without user intervention.

The supplicant is based upon the open source supplicant (albeit an earlier revision that matches that used by the NetBSD distribution) located at http://hostap.epitest.fi/wpa_supplicant/.

The supplicant provides a number of key features to support wireless connectivity. These are:

- Provides a consistent interface for configuring all authentication / encryption mechanisms (unsecure, wep, WPA, WPA2)
- Supports configuration of adhoc and infrastructure modes of operation
- Maintains the network configuration information in a configuration file (by default /etc/wpa_supplicant.conf)
- Provides auto-connectivity capability allowing a client to connect into a WAP without user intervention

An example wpa_supplicant.conf file is installed in \$STAGE/etc for you. It contains a detailed description of the basic supplicant configuration parameters and network parameter descriptions (and there are lots of them) and example network configuration blocks.

In conjunction with the supplicant is a command line configuration tool called *wpa_cli*. This tool lets you query the stack for information on wireless networks as well as update the configuration file on the fly. We are also in the process of developing a library of routines that will be pulled into a GUI (or that you can use yourself to create a Wi-Fi configuration tool). This library can be found under the source tree in lib/wlconfig and creates a libwlconfig library for applications to use.

The following commands are currently supported by the wpa_cli utility:

```
status [verbose] = get current WPA/EAPOL/EAP status
mib = get MIB variables (dot1x, dot11)
help = show this usage help
interface [ifname] = show interfaces/select interface
level <debug level> = change debug level
license = show full wpa_cli license
logoff = IEEE 802.1X EAPOL state machine logoff
logon = IEEE 802.1X EAPOL state machine logon
set = set variables (shows list of variables when run without arguments)
pmksa = show PMKSA cache
reassociate = force reassociation
reconfigure = force wpa_supplicant to re-read its configuration file
preauthenticate <BSSID> = force preauthentication
identity <network id> <identity> = configure identity for an SSID
password <network id> <password> = configure password for an SSID
new_password <network id> <password> = change password for an SSID
pin <network id> <pin> = configure pin for an SSID
otp <network id> <password> = configure one-time-password for an SSID
passphrase <network id> <passphrase> = configure private key passphrase
for an SSID
bssid <network id> <BSSID> = set preferred BSSID for an SSID
list_networks = list configured networks
select_network <network id> = select a network (disable others)
enable_network <network id> = enable a network
disable_network <network id> = disable a network
add_network = add a network
remove_network <network id> = remove a network
```

set_network <network id> <variable> <value> = set network variables (shows list of variables when run without arguments)
get_network <network id> <variable> = get network variables
save_config = save the current configuration
disconnect = disconnect and wait for reassociate command before connecting
scan = request new BSS scan
scan_results = get latest scan results
get_capability <eap/pairwise/group/key_mgmt/proto/auth_alg> = get capabilities
terminate = terminate wpa_supplicant
quit = exit wpa_cli

If you want the wpa_cli to be capable of updating the wpa_supplicant.conf file, edit the wpa_supplicant.conf file and uncomment the "update_config=1" option. (Note that when wpa_cli re-writes the configuration file, all of the comments in there will be stripped out.) Copy the file into /etc (and make sure that it's owned and read/writable by root only. It contains clear text keys and password information).

Given a system with a USB-[WiFi](#) dongle based on the RAL chips, here's a sample session showing how to get things working with a WEP based WAP.

```
# cp $HOME/stage/etc/wpa_supplicant.conf /etc
# chown root:root /etc/wpa_supplicant.conf
# chmod 600 /etc/wpa_supplicant.conf
# io-pkt-v4-hc -dural
# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
    inet 127.0.0.1 netmask 0xff000000
ural0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    ssid ""
    powersave off
    address: 00:ab:cd:ef:d7:ac
    media: IEEE802.11 autoselect
    status: no network
# wpa_supplicant -B -iural0
# wpa_cli
wpa_cli v0.4.9
Copyright (c) 2004-2005, Jouni Malinen <jkmaline@cc.hut.fi> 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.

Selected interface 'ural0'

Interactive mode

```
> scan
OK
> scan_results
bssid / frequency / signal level / flags / ssid
00:02:34:45:65:76  2437  10 [WPA-EAP-CCMP] A_NET
00:23:44:44:55:66  2412  10 [WPA-PSK-CCMP] AN_OTHERNET
00:12:4c:56:a7:8c  2412  10 [WEP] MY_NET
> list_networks
network id / ssid / bssid / flags
```

```
0 simple any
1 second ssid any
2 example any
> remove_network 0
OK
> remove_network 1
OK
> remove_network 2
OK
> add_network
0
> set_network 0 ssid "MY_NET"
OK
> set_network 0 key_mgmt NONE
OK
> set_network 0 wep_key0 "My_Net_Key234"
OK
> enable_network 0
OK
> save
OK
> list_network
network id / ssid / bssid / flags
0 QWA_NET any
> status
<2>Trying to associate with 00:12:4c:56:a7:8c (SSID='MY_NET' freq=2412 MHz)
<2>Trying to associate with 00:12:4c:56:a7:8c (SSID='MY_NET' freq=2412 MHz)
wpa_state=ASSOCIATING
> status
<2>Trying to associate with 00:12:4c:56:a7:8c (SSID='MY_NET' freq=2462 MHz)
<2>Associated with 00:12:4c:56:a7:8c
<2>CTRL-EVENT-CONNECTED - Connection to 00:12:4c:56:a7:8c completed (auth)
bssid=00:12:4c:56:a7:8c
ssid=MY_NET
pairwise_cipher=WEP-104
group_cipher=WEP-104
key_mgmt=NONE
wpa_state=COMPLETED
> quit
# dhcp.client -i ural0
# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
    inet 127.0.0.1 netmask 0xff000000
ural0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ssid MY_NET nwkey My_Net_Key234
    powersave off
    bssid 00:12:4c:56:a7:8c chan 11
    address: 00:ab:cd:ef:d7:ac
    media: IEEE802.11 autoselect (OFDM54 mode 11g)
    status: active
    inet 10.42.161.233 netmask 0xfffffc00 broadcast 10.42.160.252
#
```