Wi-Fi Protected Setup in the wpa_supplicant

The goal of Wi-Fi Protected Setup (WPS) is to automate the creation of a secure wireless network. The protocol removes the need for users to understand what a SSID is or the difference between WEP, WPA, and WPA2 and their associated ciphers. Essentially, the protocol automatically creates the network blocks specified by the wpa_supplicant's configuration file. As such, the user typically only needs to run the WPS protocol once for each AP. In WPS's simplest topology, there are two participants: the registrar and the enrollee. The registrar has the authority to issue and revoke credentials on the network. Typically, but not always, this is the access point. The enrollee, on the other hand, is the device seeking to join the wireless network. The purpose of the WPS patch is to add the enrollee function to the wpa_supplicant. This document describes how to setup the supplicant either via the configuration file or the control interface to use WPS.

WPS Methods

Before you can configure the supplicant, it is helpful to understand the two supported configuration methods: Push Button Control (PBC) and PIN. Both methods accomplish the same goal, but provide different levels security and ease of use. The PBC method is the simpler of the two. In this method, the user pushes a button on the registrar (usually the AP) and a button on the enrollee (a laptop, cell phone, etc). The protocol then takes care of connecting to the correct AP and retrieving the encryption settings. The advantage of PBC is a very simple user interface, but there are a few issues:

- The user must push both buttons within two minutes of each other. WPS refers to this time period as the "walk time".
- Only one enrollee can use the PBC method at a time. The second enrollee using PBC will receive an error message and can either elect to wait until the other enrollee is done or use the PIN method without waiting. Note that this restriction is across all detected SSID and BSSID on all channels.
- The method is unauthenticated and does not protect against active attacks

The PIN method requires the user to retrieve a PIN number from the enrollee and enter it into the registrar either via a key pad or, more likely, through a web based interface. The user then notifies the enrollee that the registrar has accepted the PIN and can proceed with connection. This method is slightly more involved than PBC, but is no more complicated than getting money from a bank's cash machine.

Supplicant Configuration

The configuration file for either PBC or PIN is the same. It should contain a network block with two elements:

```
network={
      proto=WPS
      eap=WPS
```

```
}
```

That's it! The differentiation between PBC and PIN is in the commands. To activate the PBC method, use the "pbc" command in wpa_cli. For example,

```
# wpa_cli pbc
OK
#
```

To activate the PIN method, use the "pin_get" command in wpa_cli to retrieve the automatically generated PIN number from the supplicant, enter the PIN into the AP (registrar), and finally use the "pin_entered" command in wpa_cli to start the connection process.

```
# wpa_cli pin_get
Selected interface 'wifi0'
15039545
<enter PIN into registrar>
# wpa_cli pin_entered
Selected interface 'wifi0'
OK
#
```

Note that the results of the configuration process exist in memory. To use this configuration information for subsequent connection, be sure to save the configuration (e.g. "wpa_cli save_config").

Control Interface

The expectation is most users will not issue the WPS commands via the command line and instead will use an application similar to Network Manager. Thus the patch adds a number of events and commands accessible via the supplicant's control interface.

Event/Command	Description
CTRL-EVENT-WPS-ERROR	Sent if the WPS method detects an error during any stage of the protocol. Valid for both PBC and PIN.
CTRL-EVENT-WPS-OVERLAP	Sent if the PBC method encounters more than one device attempting to use the PBC method. Not valid for the PIN method.
CTRL-EVENT-WPS-SUCCESS	Sent if the current method succeeds. Valid for both PBC and PIN. After generating this event, the supplicant will de-authenticate and associate with the new credentials. Note that some AP will change the value of the SSID over the course of WPS configuration the subsequent use of WPA*.

PBC	Starts the PBC method and initiates a "walk time" timer. The command generates one of the WPS events (error, overlap, success). Not used with the PIN method.
PIN_GET	Generates a new 8 digit PIN number each time the command executes. Not used with the PBC method
PIN_ENTERED	Starts the PIN method and initiates a 120 second timer. The command generates either the error or success WPS events. Not used with the PBC method.

WPS uses EAP to send protocol messages, but maps the protocol into a custom EAP method. This is unimportant to the user, but developers should be aware of this detail because the control interface will receive several EAP related events. Below is an example of PBC. One item that should stick out is the "EAP authentication failed" message. In WPS, EAP-Failure indicates the end of both successful and unsuccessful registrations. In this case, the subsequent CTRL-EVENT-WPS-SUCCESS indicates a successful registration. The supplicant then disconnects and associates using WPA with TKIP. Don't forget to save this configuration before quitting!

```
# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
update_config=1
network={
        proto=WPS
        eap=WPS
}
# wpa_cli
wpa_cli v0.5.10
Copyright (c) 2004-2008, Jouni Malinen <j@w1.fi> and contributors
Interactive mode
> pbc
OK
<2>Trying to associate with 00:1c:f0:ff:6a:9e (SSID='dlink6A9E' freq=5805 MHz)
<2>CTRL-EVENT-DISCONNECTED - Disconnect event - remove keys
<2>CTRL-EVENT-DISCONNECTED - Disconnect event - remove keys
<2>Associated with 00:1c:f0:ff:6a:9e
<2>CTRL-EVENT-EAP-STARTED EAP authentication started
<2>CTRL-EVENT-EAP-METHOD EAP vendor 14122 method 1 (WPS) selected
<2>CTRL-EVENT-EAP-FAILURE EAP authentication failed
<2>CTRL-EVENT-WPS-SUCCESS
<2>CTRL-EVENT-DISCONNECTED - Disconnect event - remove keys
<2>Trying to associate with 00:1c:f0:ff:6a:9e (SSID='dlink6A9E' freq=5805 MHz)
<2>CTRL-EVENT-DISCONNECTED - Disconnect event - remove keys
<2>CTRL-EVENT-DISCONNECTED - Disconnect event - remove keys
<2>Associated with 00:1c:f0:ff:6a:9e
<2>WPA: Key negotiation completed with 00:1c:f0:ff:6a:9e [PTK=TKIP GTK=TKIP]
<2>CTRL-EVENT-CONNECTED - Connection to 00:1c:f0:ff:6a:9e completed (auth)
[id=0 id_str=]
> save_config
OK
```

```
> quit
# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
update_config=1
network={
        ssid="dlink6A9E"
        psk=62edba2fcae92265da5414fc967c4bf5c62963cca09385cf699957d9f66a0586
        proto=WPA
        key_mgmt=WPA-PSK
        auth_alg=OPEN
        eap=WPS
}
#
```

Including WPS in the Build

The in-band WPS EAP registration methods exclusively use the internal cryptographic functions. Therefore, be sure to include

```
CONFIG_IEEE8021X_EAPOL=y
CONFIG_TLS=internal
CONFIG_EAP_WPS=y
```

in the .config file.